How to Generate Supernova Light Curves: Photometry with the Katzman Automatic Imaging Telescope (KAIT) and Its Application to Type Ia Supernovae, Cosmology, and the Accelerating Universe.

Andrew Friedman [1,2]

June 11, 2002

ABSTRACT

This primer is intended to be a guide for students hoping to generate supernova light curves with data from images taken by the Katzman Automatic Imaging Telescope (KAIT), although some of the techniques discussed will be applicable towards data reduction in general. Generating these light curves involves learning how to use the Unix Operating System, learning how to do Photometry in IRAF (Image Reduction and Analysis Facility), and possibly learning some computer programming in IDL (Interactive Data Language), a programming language specifically designed for image processing in astronomy. Many of the supernovae we have data for were found by undergraduates in this group through the Lick Observatory Supernova Search project (LOSS), of which most of you have probably already been a part. Other supernovae were discovered by other research groups, after which follow up observations were taken with KAIT. The basic idea is that a supernova's light curves plot its apparent brightness as a function of time in several different filters (U, B, V, R and I bands), which let in light only from within a specific wavelength range. In general, a supernova explodes, brightens to some maximum in each band in roughly 2 weeks, and then fades over the course of roughly 6 months to a year. The shapes and peak brightnesses of these light curves in different bands can be very sensitive to particular cosmological models, and thus can be used to place sharp observational constraints on cosmology. The big picture goal of the project is thus to discover large numbers of these supernovae in distant galaxies, (specifically Type Ia supernovae) determine their spectra and light curves from follow up observations, and use them to do cosmology. Specifically, we hope to measure the acceleration rate of the expansion of the universe. At this point, if you're reading this document and preparing to do Photometry on a supernovae, you are already at the final step in the process: The supernova has been discovered, spectra were taken, (perhaps at Keck) to determine if the supernova was a Type Ia, follow up observations have been taken by KAIT and all that remains is to generate the supernova's light curves. This is where you come in, and since the task certainly can appear daunting at first, hopefully, this primer will help make it a bit easier.

---

[1] e-mail: andyf365@uclink4.berkeley.edu (soon to be afriedman@cfa.harvard.edu)

[2] The Lick Observatory Supernova Search Group (LOSS) headed by Professor Alex Filippenko, Astronomy Department, University of California, Berkeley, Berkeley, CA, 94720. This document created with LaTeX 2_ε

Contents

1. What Are We Doing and Why Are We Doing It in the First Place?

Personally, I have been doing research here at Berkeley as a member of the Lick Observatory Supernova Search (L.O.S.S) with Professor Alex Filippenko and Assistant Research Astronomer Weidong Li since January of 1999. My research in particular has focused on both searching for supernovae (which most of you probably have already done) and performing photometry on the follow up observations to obtain the supernova's light curves (which you are learning to do). I am motivated to write this primer as a sort of legacy to leave to the future generations of undergrads and grad students in the group who want to do photometry and, at the same time, understand how the work fits into the big picture of astrophysics, cosmology, and the accelerating universe results.

1.1. Warnings and Basic Assumptions of this Primer

This primer is intended for both students who just want to do photometry, and also for students who are interested in the big picture motivation behind it all. Thus section 2 is optional if you just want to get into the photometry immediately. However, I suspect that having the relevant background knowledge regarding cosmology and the accelerating universe results will prove invaluable. But in any case, if you feel sufficiently motivated already, and just want to learn the skills, then feel free to read section 1 and then skip to section 3. The sections marked in the table of contents (and throughout) with an asterisk * may or may not be needed, but please read the beginning of those sections to be sure. That said, I'll continue with some warnings about the photometry process itself.

Unfortunately, I must warn you that it is easy enough to go through the photometry process and produce your light curves mechanically without understanding any of the fundamental motivations behind it all. To be perfectly honest, in undertaking this research project, you could easily spend all of your time in front of a computer, (a Sun workstation in 705 or 727 Campbell, in this case). You could, without difficulty, perform photometry, for example, without ever going out to the telescope, (although it would be a very reasonable thing to do for independent reasons). You will certainly need to learn Unix and IRAF, and you may learn to program in IDL, but you could also end up primarily using IRAF software or software written by Weidong Li, without writing any of the code yourself or having any practical need to understand what the programs are doing at the fundamental level. It is exceedingly clear that you have to understand a program much more if you write it yourself. If you use other people's software, you may be many levels divorced from reality, and the practical danger is that without knowing what is going on, it is easy to become essentially a manual task robot. In this situation, it not hard to become dissatisfied with the work and to ultimately question whether this is what you want to do with your time as an undergrad (or grad student). Thus I hope to explain as clearly as possible, what is going on at each step, to help you understand what you're doing, and allow you to not lose sight of the big picture.

I also will try not to take too much for granted, regarding the details of what you have learned thus far. I won't assume, for example that you have had experience with Unix, IRAF, IDL, or any programming whatsoever. Although I will certainly have to make some assumptions for the sake of brevity. I will assume, for example, some familiarity with basic physics, chemistry, math, and astronomy, i.e. electrons, photons, the atomic elements, basic statistics, errors, astronomical magnitudes, apparent brightness vs. intrinsic luminosity, stars, telescopes, redshift, spectral lines, the cosmic microwave background, cosmological distance indicators, Cepheid variable stars, thermonuclear fusion in stars, basic cosmological models and that sort of thing. Essentially this is at the level of Astro 7A-B, which is probably a good background to have before starting photometry.

SUPERNOVA GALLERY

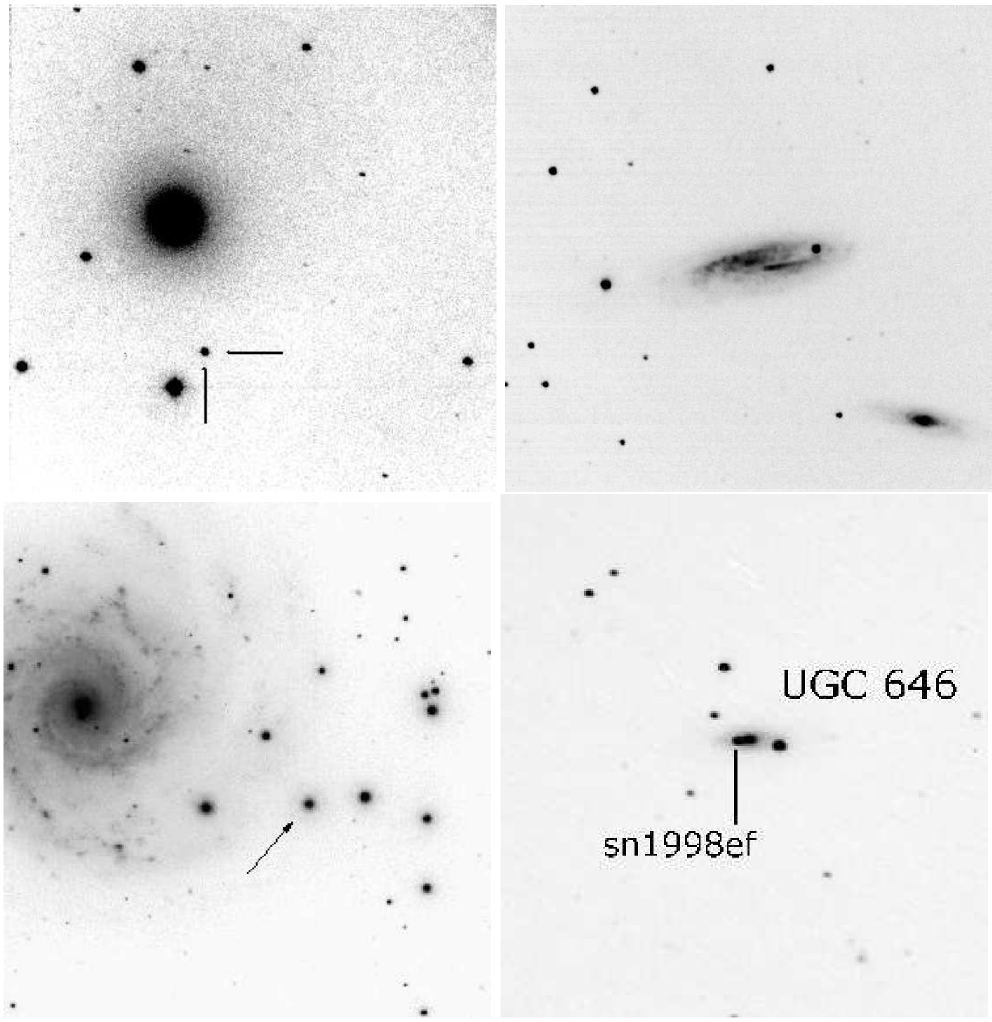

Fig. 1.— Supernova Gallery. From left to right, we have sn2000cx, sn1998dh (upper right corner of galaxy), sn2002ap, and sn1998ef. These are all supernovae that I've had a chance to do work on to varying degrees. I did photometry on 2000cx and 1998dh, helped a little on 2002ap, and as I am writing this primer, am working on 1998ef, but probably will leave it for others to finish.

1.2. Data Reduction: From CCD Images to Standard Light Curves

To begin, I want to present an idea in advance of the starting and endpoints of the project, to give you a tangible goal to look toward, since its easy enough to get lost and completely forget the big picture. Basically, when you begin the photometry process, you start with a data set of optical CCD images of follow up observations taken by KAIT after the supernova was discovered. You can see a sample data set of B band images for supernova 1998dh below.

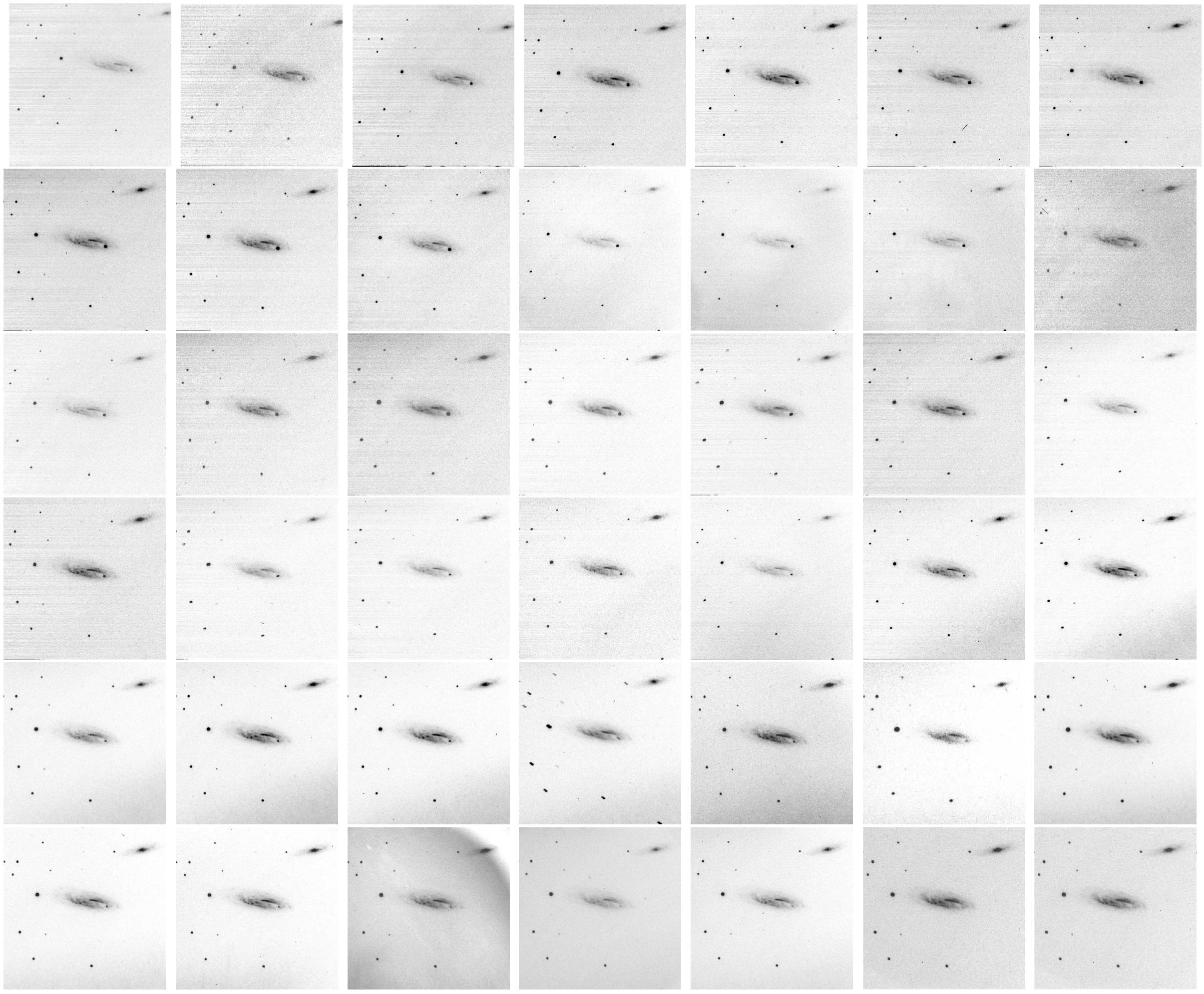

Fig. 2.— 40 unreduced, B band images for supernova 1998dh. The Images are displayed in time order, with observation dates ranging from 7/25/1998 to 12/18/1998. The supernova itself is a tiny black dot located in the lower right corner of the host galaxy and can be seen to brighten to a maximum at around the 10th day and then fade. The black parts in the images are the brightest. The changes in contrast in the images are sometimes due to the weather and sometimes due to the scaling with which the image was saved. The final 2 images are Galaxy Template Images taken well after the supernova has become too faint to see (9/25/2000 and 7/19/2001).

Before you do anything to this data, the data is called "unreduced" or "raw" data. After you do photometry, and "reduce" the data, your final product will be a set of light curves in each band, where you measure the brightness of the supernova in each image in each band (in units of astronomical magnitudes) and plot it vs. the date and time of observation for that image (in units of Julian days).[3] Basically, light curves plot the brightness of the supernova as a function of time.

From your data set, you first generate Instrumental light curves, which are dependent on the instruments you used, i.e. your CCD camera, the optics of your telescope, the filters you used for each of the wavelength bands[4], your software, etc.. From this, you can get at most a sense of the relative behavior of the supernova in each of the bands. The standard light curves, however, take the instrumental light curves and calibrate them by putting them on a standard scale that all astronomers can agree upon. So if you say that supernova 1998dh reached a peak B band magnitude of 14.11 $\pm$ 0.01 on Julian day 2451029, then astronomers know what you are talking about and can view that information as observer independent to within the error bars. The plot of your standard light curves, along with a tabular list of your data and the associated errors for each data point, are the end products of photometry that you hope to present in publishable form.

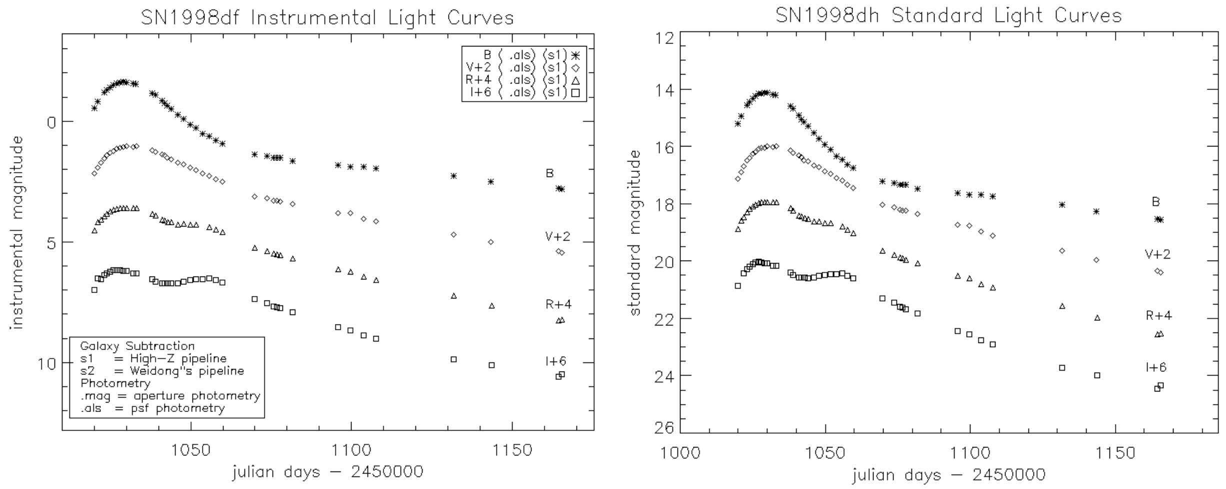

Fig. 3.— Instrumental and Standard $BVRI$ Light Curves for supernova 1998dh. This is the major graphical form in which you eventually would publish this data.

---

[3]The Julian calendar, established by Julius Caesar in the year 45 BCE, was corrected by Pope Gregory XIII in 1582, excising ten days from the calendar. (Taken from IDL help ?Julian) For example, the Julian date for December 20, 1998 at Universal Time (or Greenwich Mean Time) 6:41:34 is 2451106.7789. As such, it is often common for astronomers to plot the time axis in Julian days - 2450000 in order to get more convenient numbers.

[4]The standard wavelength bands for optical telescopes are U, B, V, R and I, which stand for Ultraviolet, Blue, Visible, Red, and Indigo respectively. By contrast, an Infrared camera has bands named H, J and K. When you view an astronomical object in a given band, you basically put a color plate over the optics that ideally lets in photons only from the desired wavelength range, but of course, only does so approximately.

1.3. Table of Photometric Data For Supernova 1998dh

Julian Date - 2450000	B	δB	V	δV	R	δR	I	δI
1020.00	15.21	0.02	15.13	0.01	14.90	0.02	14.87	0.02
1021.00			14.71	0.02	14.46	0.01	14.44	0.03
1023.00	14.56	0.01	14.50	0.02	14.31	0.01	14.27	0.03
1024.00	14.44	0.02	14.39	0.01	14.20	0.01	14.18	0.01
1025.00	14.33	0.02	14.27	0.02	14.09	0.02	14.11	0.02
1026.00	14.25	0.02	14.19	0.01	14.03	0.02	14.04	0.02
1027.00	14.17	0.02	14.09	0.01	13.98	0.01	14.03	0.01
1028.00	14.14	0.02	14.07	0.02	13.96	0.01	14.05	0.02
1029.00	14.12	0.01	14.05	0.02	13.95	0.01	14.08	0.03
1030.00	14.14	0.02	14.01	0.02	13.94	0.01	14.08	0.03
1032.00	14.18	0.04	14.03	0.02	13.95	0.01	14.16	0.02
1032.75	14.23	0.02	14.00	0.02	13.96	0.03	14.17	0.04
1038.00	14.60	0.05	14.15	0.04	14.17	0.04	14.41	0.05
1039.00	14.70	0.03	14.23	0.04	14.25	0.03	14.48	0.03
1041.00	14.92	0.03	14.32	0.03	14.40	0.02	14.57	0.03
1041.75	15.05	0.03	14.37	0.03	14.45	0.02	14.58	0.04
1042.75	15.16	0.02	14.48	0.02	14.52	0.01	14.57	0.01
1044.00	15.29	0.04	14.52	0.04	14.53	0.02	14.59	0.03
1046.00	15.52	0.03	14.68	0.02	14.63	0.02	14.59	0.03
1047.75	15.73	0.02	14.74	0.01	14.62	0.01	14.53	0.03

Table 1: Table of final photometric $BVRI$ data and errors for supernova 1998dh. This, along with the plot of the same data is the final publishable output of your work. In general, you don't see the error bars on the previously shown light curve plots because they are so small that they would actually be smaller on the y-axis than the plot symbol for each band (i.e. asterisk, circle, triangle, square) on the scale of the light curve plot shown before. This is because the accuracy of photometry as above is often on the order of 0.1%

Table of Photometric Data For Supernova 1998dh (continued...)

Julian Date - 2450000	B	δB	V	δV	R	δR	I	δI
1049.75	15.95	0.01	14.88	0.02	14.67	0.01	14.49	0.03
1051.75	16.12	0.05	14.97	0.03	14.69	0.03	14.46	0.05
1053.75	16.35	0.02	15.10	0.02			14.46	0.02
1055.75	16.47	0.04	15.20	0.02	14.81	0.01	14.44	0.03
1057.75	16.64	0.02	15.32	0.02	14.90	0.01	14.53	0.03
1059.75	16.76	0.02	15.45	0.02	15.02	0.02	14.61	0.02
1069.75	17.22	0.02	16.05	0.01	15.65	0.02	15.30	0.02
1073.75	17.29	0.01	16.13	0.01	15.80	0.01	15.45	0.01
1075.75	17.34	0.01	16.22	0.02	15.88	0.03	15.61	0.02
1076.75	17.34	0.02	16.23	0.03	15.91	0.02	15.62	0.03
1077.75	17.35	0.01	16.26	0.02	15.95	0.03	15.67	0.04
1081.75	17.49	0.02	16.36	0.01	16.07	0.01	15.83	0.01
1095.75	17.62	0.04	16.74	0.03	16.51	0.03	16.44	0.03
1099.75	17.70	0.03	16.76	0.01	16.60	0.00	16.56	0.02
1103.75	17.69	0.02	16.98	0.02	16.82	0.02	16.77	0.04
1107.75	17.73	0.03	17.11	0.02	16.94	0.02	16.89	0.04
1131.75	18.04	0.03	17.65	0.04	17.57	0.06	17.74	0.05
1143.50	18.28	0.03	17.97	0.03	17.97	0.05	17.98	0.05
1164.50	18.55	0.03	18.35	0.03	18.55	0.03	18.46	0.03
1165.50	18.57	0.03	18.40	0.03	18.54	0.04	18.34	0.04

Table 2: Table continued...Table of final photometric $BVRI$ data and errors for supernova 1998dh. This, along with the plot of the same data is the final publishable output of your work. In general, you don't see the error bars on the previously shown light curve plots because they are so small that they would actually be smaller on the y-axis than the plot symbol for each band (i.e. asterisk, circle, triangle, square) on the scale of the light curve plot shown before. This is because the accuracy of photometry as above is often on the order of 0.1%

2. The Big Picture: Cosmology and the Accelerating Universe

So now that you know what a light curve is, I can get onto answering the question of why were making them in the first place. What are they good for anyway? Thus, in the next few sections in particular, I hope to convey, at the very least, a sense of the scientific reasons for undertaking this particular project. And at least in a qualitative sense, I hope to explain how your work fits into the larger picture of cosmology and the accelerating universe results, and how it may help make an impact on furthering our understanding of astrophysics. Personally, I can not do science in a vacuum, so I always find it useful to keep in mind why I am doing what I am doing. A scientist who loses track of the big picture is simply doing themselves and their peers a disservice.

2.1. The Cosmological Constant Problem

That said, our particular LOSS group works under the larger auspices of the High Z Supernova Search Team, headed by Dr. Brian Schmidt at the Australian National University in Canberra. High Z means high redshift (roughly $z > 0.1$). In the LOSS group, we look for nearby, low redshift supernovae ($z < 0.1$), and compare and combine our data with the data from the High Z Team (HZT). The big picture goal of the project is to discover large numbers of Type Ia supernovae in nearby and distant galaxies, determine their spectra and light curves from follow up observations, and use them to do cosmology. These light curves are then used to test various cosmological models, and specifically, to measure the acceleration rate of the universe. Since Edwin Hubble's 1929 discovery that almost all galaxies are redshifted, and are thus receding from us, we have known that the universe is expanding. The major result of the High Z Team, which was announced first in 1998, is that the supernovae we observed are consistently 10-15% dimmer than what we would expect from a coasting expanding universe, implying that they were actually farther away than expected, and thus that the expansion of the universe may actually be accelerating.

When I joined the group in early 1999, the result was even more controversial than it is now because it flew in the face of conventional wisdom that the universe was surely decelerating, as gravity should eventually overcome the energy from the initial expansion. The results also resurrected serious interest in Einstein's Cosmological Constant Λ, which in theory could provide a cosmic antigravity force, which resists normal gravity and causes the acceleration. As it happened, the accelerating universe results were obtained independently through work done by the Supernova Cosmology Project (SCP) headed by Dr. Saul Perlmutter at the Lawrence Berkeley National Laboratories. This gave both of our groups more confidence in a result that initially, neither team was willing to believe, let alone publicize to the astronomical community. Since then, crucial independent tests of the possible cosmological models have been performed using data from the Cosmic Microwave Background (CMB), most notably from the MAXIMA, BOOMERanG, and DASI experiments, along with longstanding galaxy cluster measurements. These results are also consistent with the supernovae results. Thus the present consensus from the astronomical community is that we have an accelerating universe with a nonzero cosmological constant, and an unwieldy list of questions regarding the true physical basis for this phenomenon.

Indeed, the idea of cosmic antigravity has been controversial enough throughout to force both the HZT and the SCP groups to become our own harshest critics, and act quite cautiously in regard to taking the results on face value. As such, we have tested for several major systematic errors, including chemical evolution of the Type Ia progenitor systems, and reddening by interstellar dust, and found that our results are largely unaffected. And now that we have independent CMB and galaxy cluster data which is consistent with the supernovae results, the astronomical community largely accepts that the universe is accelerating. The supernova results are far from dogma, but they are beginning to be incorporated into the curriculum of advanced astrophysics courses and newer textbooks, for example. In the cautious spirit of the scientists we have been fortunate to learn from, we must thus critically ask the question, ***"Is there any other effect which could be mimicking what we interpret to be cosmological acceleration?"*** In other words, are these results physical, are they due to flawed data analysis, or are they simply an artifact of an incorrect interpretation stemming from an incomplete understanding of fundamental physics?

First of all, disregarding for a moment my personal bias, it is not unreasonable to speak of the accelerating universe results as one of the most important recent discoveries in all of physics, and, quite possibly, all of science. It is not inconceivable that, if the results hold up, the leaders of the relevant projects (Brian Schmidt, Saul Perlmutter, and Adam Riess ?) could merit a shared Nobel Prize in physics within the next decade. When a scientific result this ground-breaking is released, an honest scientist thus has to meet the idea with a reasonable amount of skepticism. We are in this field to answer fundamental questions about the past, present, and future state of the universe as a whole, and these cosmological questions are philosophically and scientifically important enough that we owe it to ourselves and the scientific community to subject any major results to sufficient peer reviewed criticism. And beyond that, it is not unreasonable to say that we actually owe it to humanity to make a sincere effort at critical examination.

Granted, most citizens of earth might value esoteric astronomical discoveries, such as, say, measuring the neutrino mass, with comparatively low regard, but for a publicly accessible result such as one that says *"The universe is roughly 14-15 billion years old."* or, *"The universe is accelerating in its expansion."*, a large number of people are actually sincerely interested and affected. Regardless of the esoteric foundations of the supernovae results, the idea of the expanding and accelerating universe happens to be quite conducive to the public imagination. It is within this big picture context that I personally am motivated to help investigate the validity of this result, aside from its pure scientific interest. If the universe is accelerating in its expansion, or if it is simply coasting along, these are reasonable things for an educated human being to be aware of, just as it is reasonable to understand that the universe likely began a finite time in the past at the Big Bang.

But just as it is reasonable to continue subjecting the Big Bang model to stricter and stricter tests, this result, as a comparative toddler, should in no way be spared from a similar gauntlet of critiques. First of all, there are several motivations for the idea that, while the data analysis underlying the supernova results is probably robust, the assumptions that the various research groups have adopted in interpreting the results may be in need of revision. The idea that the data analysis may be robust comes from the fact that it is statistically unlikely that the independent data analyses done by the High Z Team, the Supernova Cosmology Project, the galaxy cluster people, and the MAXIMA, BOOMERanG, and DASI, CMB collaborations are all intrinsically flawed, yet somehow still in agreement. But if the shared theoretical assumptions made by the all the groups are flawed at some level, the interpretation of the results would be in question, even if we assume perfect data analysis. The assumptions stem from our current understanding of Einstein's theory of General Relativity.

In the current interpretation, the acceleration is caused by Einstein's newly resurrected cosmological constant Λ, which was originally an ad hoc repulsive gravitational term thrown in to Einstein's Field Equations in order to keep the universe static, as Einstein believed it to be at the time. He later retracted the idea when Hubble found the universe to be expanding and called the introduction of Λ "The biggest blunder of his career." As it happens, Einstein may not have been wrong after all, as Λ now serves as a repulsive gravitational force (cosmic antigravity) which causes the expansion of the universe to accelerate. Even so, The idea of the cosmological constant being a real, physical quantity has left a large number of scientists uneasy for several reasons. When we apply standard particle physics theory, which treats the natural Planck energy density of the vacuum as the physical basis for the cosmological constant, Λ, we get a number that is 10^{120} orders of magnitude larger than what we observe, quite possibly the most numerically discrepant prediction in the history of science! If the independent supernovae and CMB results are as robust as they seem, this clearly signifies that we are likely to require significant change in our understanding of fundamental physics in order to describe what, in fact, Λ actually is. As it stands, our current fundamental physics gives us an answer that is bordering on ridiculous.

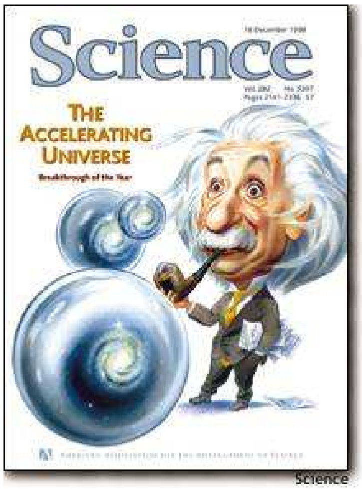

Fig. 4.— Science Magazine named the discovery of the Accelerating Universe the top science breakthrough of 1998. Thus we see why Einstein's caricature above is pleasantly surprised at the possibility that Λ might not have been a blunder after all. The sheer excitement of this result was undoubtedly one of the reasons I joined the Filippenko group in early 1999.

This alone naturally leads one to consider that maybe we are being fooled somehow, observing an effect that looks like Λ and cosmological acceleration, but is in fact, some other new physical phenomenon altogether. Granted, this would defer the problem of what Λ is to some other source, but if the other source is more amenable to and consistent with theoretical and observational tests, then as good scientists, we can learn to live without Λ. Λ it is still our best bet thus far, but as honest scientists, we must continue to be our own harshest critics and seriously consider the possibility that maybe General Relativity and the Standard Model of Particle Physics are flawed (one of them almost certainly is), and that the Λ we think we are seeing is actually an artifact of some other yet to be understood physical process. What that process might be is anybody's guess, but this is what makes the cosmological constant problem such an interesting conundrum. The point is that the question of whether Λ is real is of great scientific interest, regardless of what the result may be. In other words, even if the idea of the cosmological constant and the accelerating universe ends up being terribly wrong, we can not help but learn some interesting physics through its investigation. And by helping to generate supernova light curves, you are playing an integral part in carrying out that investigation.

2.2. The Theory of Type Ia Supernovae

Now that we've described some of the relevant cosmological implications that might motivate you to churn out some supernova light curves, lets discuss a bit more about some of the physical details of how we use supernovae for cosmology, focusing on Type Ia supernovae in particular. Basically, to use supernovae to answer cosmological questions, we first look at its spectra to determine its redshift and see if we can classify it as a Type Ia. A Type Ia spectrum is characterized by the lack of Hydrogen lines and the presence of strong Si II absorption. On the other hand, a Type II supernovae, for example, shows Hydrogen in its spectrum. There are many more spectral based sub-classifications of supernovae (Type Ib, Type IIn, Type Ic etc...) that we need not go into detail about here, but in general, we believe that Type II supernovae are due to the explosion of massive stars ($M > 10M_{\odot}$) in Spiral Galaxies and Type Ia supernovae are due to the thermonuclear explosion of White Dwarf stars in either Spiral or Elliptical galaxies. These conclusions are based on strong empirical evidence from astronomical observations and further theoretical evidence which involves modeling the spectra and light curves of supernovae based on the chemical composition of the exploding star and the elements it ejects upon explosion. Even so, Type II supernovae, which I will not discuss here, are much better understood than Type Ia supernovae, and we must recognize this if we are relying on Type Ia supernovae for cosmology.

We use Type Ia supernovae in particular because there is a legitimate theoretical basis for the idea that all Type Ia supernovae can be treated approximately as "standard candles", in the sense that their intrinsic brightness does not vary much amongst populations of Type Ia's exploding over cosmological timescales and thus at different redshifts. This assumption is based on the idea that Type Ia supernovae occur due to the thermonuclear disruption of White Dwarf stars near the Chandrasekhar mass limit of roughly $M_{ch} = 1.4M_{\odot}$. Since these White Dwarfs are thought to explode at roughly the same mass, and since mass is, to first approximation, the major factor determining the luminosity of the explosion, it is thus reasonable to conclude that the intrinsic brightness of these explosions should be roughly uniform. Assuming that Type Ia supernovae are good standard candles, we can use them as relatively robust cosmological distance indicators. Traditionally, we get a distance by measuring the supernova's apparent peak brightness from its observed light curve, and comparing it with that of other Type Ia's that exploded in galaxies whose distance we have determined independently, for example using Cepheid variable stars. By comparing Type Ia's from a distant high redshift galaxy with other Type Ia's from a nearby low redshift galaxy with a known distance, we can then deduce the distance to the high redshift galaxy and place it on the cosmological distance ladder. Knowing both the distance and the intrinsic brightness of the high and low z supernova, we can compare the apparent brightness we observe for them to the apparent brightness we would expect to see from models with or without cosmological acceleration, and test which model is most consistent with observations.

However, comparison between the light curves of high and low redshift supernovae in different environments is necessary to justify the assumption that Type Ia supernovae can indeed be used as reliable standard candles. Other effects, such as evolution of the chemical composition of successive generations of these objects, or reddening by interstellar dust, could in theory change the intrinsic brightness of Type Ia supernovae as a function of redshift. Testing this assumption and providing the foundation for the understanding the high redshift supernovae is a primary focus of the particular supernova search that we are involved in. Thus, as discussed, we look for nearby, low redshift supernovae, ($z < 0.1$), whose spectra and light curves we determine and then compare and combine with those of the high redshift supernovae found by the High Z Team. By comparing both high and low redshift supernovae, we can test for chemical evolution of the progenitor systems, and extend our cosmological tests to regimes that require both high and low redshift data points.

Even so, it must be stressed that the theory of Type Ia supernovae is still really not well understood. Most physicists and astronomers agree that Type Ia's are probably due to the thermonuclear disruption of primarily Carbon-Oxygen white dwarf stars in a binary system, where the white dwarf accretes matter, from, say a red giant companion star until it reaches $M_{ch} = 1.4\ M_{\odot}$.[5] Progenitor systems do exist, but unfortunately, the mass accretion rate onto the white dwarf must be finely tuned: too slow and it doesn't accrete up to M_{ch} in the Hubble Time (i.e. the age of the observable universe), too fast and the white dwarf can easily blow off more mass than it accretes through surface nova explosions. Furthermore, it requires the white dwarf to already be close to M_{ch}, say at maybe 1.0 - 1.3 $M_{\odot}$, and these types of white dwarf stars are much rarer than the average white dwarf star of about 0.6 $M_{\odot}$. Thus it is not clear that it is easy to get up to the Chandrasekhar mass. This is the so-called "single degenerate" scenario, because a white dwarf is made of degenerate[6] matter, and there's only one of them in the binary system.

---

[5]The reason for the explosion is quantum mechanical in nature, but I do not have time to go into the details here.

[6]Degenerate in this case is a quantum mechanical term denoting a different state of matter, where all the available electron energy levels are occupied. White dwarf stars are supported against gravity by electron degeneracy pressure.

Another way to get up to the Chandrasekhar mass is in the so-called "double degenerate" scenario' where you have a "white dwarf - white dwarf" binary system where the two stars merge and the total mass exceeds M_{ch}. This scenario would naturally explain the observed lack of hydrogen in the Type Ia spectrum (no contamination from the hydrogen rich companion star), and allows you to begin with the more common types of white dwarfs that are roughly 0.6 - 0.8 $M_{\odot}$. Unfortunately, it is not clear that there are enough white dwarf binary candidates that would merge through the emission of gravitational radiation in the Hubble Time in order to make this mechanism a prominent producer of Type Ia's.

Fig. 5.— Artist's conceptions of the single degenerate and double degenerate scenario for Type Ia supernovae. Artist Credit: Dana Berry and Dan Durda.

In the end, it is probably likely that both scenarios contribute to the total population of Type Ia supernovae, each with some probability. This is possibly evidenced by the fact that, in reality, Type Ia supernovae are actually *not* perfect standard candles. There is a scatter in the peak apparent brightness of the light curves which is not consistent with all Type Ia's having the same intrinsic brightness at all redshifts. There are "subluminous" Ia's such as (sn1986G) and "overluminous" Ia's such as (sn1991T), and even "very subluminous" supernovae like (sn1991bg), so something has to give. But we do not know enough for now about the progenitors systems to say, for example, that 30% are double degenerate, and 70% are single-degenerate.

Furthermore, the explosion mechanism itself is terribly misunderstood. No one knows for sure whether the nuclear detonation wave begins in the center of the star or off center, or exactly how the blast wave proceeds. The details of this process are likely to be largely chaotic, and thus, extremely sensitive to initial conditions, especially to the initial chemical composition of the exploding white dwarf, which in most cases, is probably Carbon-Oxygen, but might also be Helium or Oxygen-Neon white dwarfs, with differing initial masses and the abundance ratios largely uncertain. Small changes could result in different amounts of heavy elements being produced in the explosion. In fact, the bulk of the energy that gets turned into visible photons that we see from the supernova, occurs due to the production of ^{56}Ni, which decays to Cobalt and emits the photons that eventually hit our CCD camera. So basically, ^{56}Ni decay powers the light curve, and the intrinsic peak brightness of the light curve in all bands will be extremely sensitive to even small changes in the amount of Nickel mass produced. From this perspective, the situation appears messier and the scatter in intrinsic brightness amongst Ia's appears almost inevitable.

However, all is not lost because there are ways to correct for the scatter by noticing statistical correlations between the shapes of many light curves, and then calibrating the data in your sample over large numbers of light curves.[7] Unfortunately, this kind of approach of applying largely empirical statistical corrections after the fact, although useful, still ignores a great deal of the fundamental physics.[8] And as scientists, we should always be wary of basing our results with shaky theoretical foundations. Nevertheless, several independent measurements of the relevant cosmological parameters using data from the Cosmic Microwave Background radiation and from Galaxy Cluster measurements, are all consistent with the supernova results.

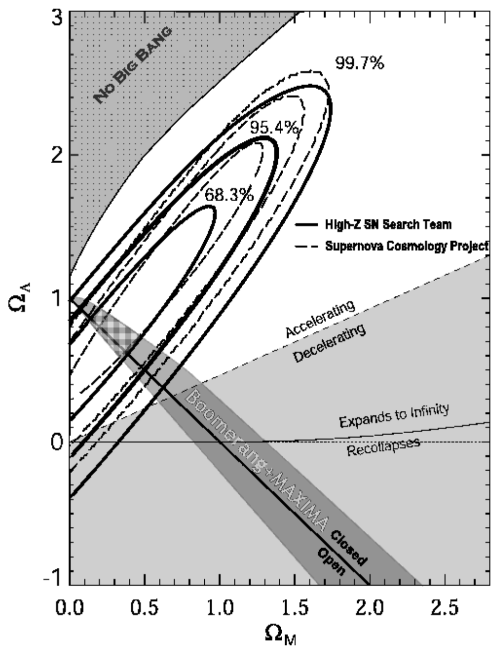

Fig. 6.— Summary of the HZT-SCP Cosmological Parameter Measurements, graphed in the Ω_M vs. Ω_λ plane. Galaxy cluster measurements also indicate independently that $\Omega_M \approx 0.3$, representing another vertical strip that happens to also intersect the combined Supernova and CMB data. Thus we see a convergence in the data from the HZT, SCP, BOOMERanG and MAXIMA collaborations at around $\Omega_M \approx 0.3$ and $\Omega_\lambda \approx 0.7$. Results from DASI are not shown. This convergence very strongly suggests that the universe is accelerating due to a nonzero cosmological constant Λ.

[7]These include Adam Riess' Multicolor Light Curve Shape (MLCS) method, and Mark Phillips' Δm_{15} method, and Saul Perlmutter's Stretch method, which all explot observed statistical correlations between many light curves.

[8]There have been some recent theoretical studies to explain the correlations physically in terms of varying amounts of Nickel mass produced. These have been relatively successful, but much work still remains to be done.

I don't have time here to do a review of cosmology, but in general, the most important cosmological parameter is Ω, which is a measure of the fractional energy density of the matter in the universe with respect to some critical reference density. With no cosmological constant, if $\Omega > 1$, the universe collapses back on itself, if $\Omega < 1$, the universe expands forever, and if $\Omega = 1$, the universe coasts and barely expands forever. If there is a cosmological constant Λ, as the supernova, CMB, and galaxy cluster results seem to indicate, then the universe accelerates in its expansion and expands forever. In this scenario, $\Omega = \Omega_M + \Omega_\lambda$, which indicate the contributions to the energy density from matter and the cosmological constant respectively. The supernova results themselves can measure the quantity $|\Omega_M - \Omega_\lambda|$. The details of exactly how we get to this quantity from the light curve are beyond the scope of this primer, but can be found in the references.

So it turns out that our empirical corrections may be valid, but not because we truly understand the physics. Thus it is useful to keep in mind all of these uncertainties when you are producing a light curve and contemplating how it will be used for cosmology. But until we understand the physics, from your standpoint, you will mainly be helping to add data to the sample, upon which we can do more of the same empirical statistics using correlations between light curve shapes, which increase in accuracy the more light curves we have and the better the data are.

That said, ultimately, we hope to generate the light curves a large number Type Ia supernovae who we have followed extensively with KAIT, but whose data are simply waiting idly by while we wait for someone to actually do the photometry and reduce the data. For many supernovae, we have to wait a long time before we can even begin doing any photometry. For supernovae requiring galaxy subtraction, we have to wait 6 months to a year after discovery to get galaxy templates after the supernova has faded, and for every supernovae, we also have to wait for a clear night to get calibration images of the standard stars, so we can transform our instrumental supernova magnitudes to standard magnitudes. And our motivation to take these steps is contingent on having people there to actually do the work.

Thus I am writing this primer primarily to help train the new people in the group, so we can efficiently approach the task of reducing the considerable about of supernova data we have. When we present the addition of the new data to our sample, the work will be of considerable interest to astrophysics, especially to those in the supernovae field, as we would effectively be extending the current useful database of nearby supernovae by roughly 30% or more. In addition, our light curves are better sampled than those previously published (i.e. more follow up images, closely spaced in time), and would likely become the new "training set", with which to compare all subsequent Type Ia supernovae light curves, providing even stricter tests of the accelerating universe results. This is the sense in which you will be contributing to cosmology at the most basic level.[9]

[9]There are many other interesting and important scientific uses for studying nearby Type Ia SNe besides cosmology, although it is the most notable application, and certainly one of the "hottest" things in modern astronomy. Aside from cosmology, applications of studies of nearby Type Ia SNe include: (1) Testing the empirical correlations, (2) Studying SNe in greater detail than the High z SNe, providing clues to the underlying physics, (3) Studying the effects from metallicity and the environment of the SNe, and (4) bulk flow measurements, amongst others.

3. Where Do You Start? Using the Unix Operating System

The primary place to go to do Photometry is at the south-most office in 727 Campbell, at the computer named "hercules", where most of you have probably already checked images for the supernova search. If you have a ugastro account[10], you can also log into hercules remotely from any of the computers in 705 Campbell, which are all part of the ugastro network. Finally, with some effort, in principle, you can even log in remotely from home if you have a fast enough connection.[11]

Hercules and all the computers in 705 use the Unix Operating system, which is another operating system just like Windows or Macintosh, only much better for astronomy and scientific research in general. It is faster, arguably much better designed, and unlike Windows, it hardly ever crashes! The only difference is that in general, it relies much less on user friendly graphical user interfaces (GUI's) like icons and folders that you double click on. Instead, you do most of your work in Unix by typing in commands on the "command line." This takes a little getting used to, but once you learn your way around on the command line, Unix is a gazillion times faster than Windows.

3.1. Logging into hercules Locally or Remotely.

You can log into hercules at the computer itself, remotely from 705 Campbell, or potentially even from home. You can log in at hercules locally simply by typing in Weidong's password.[12] The machine is always on, with the user logged in as Weidong. Typing in the password simply unlocks the screen and lets you access the desktop.

You can login to hercules remotely from 705 Campbell by first entering your ugastro username and password, and then typing in the following at the command line:

```
> slogin -l weidong hercules.berkeley.edu
```

You will then be asked to enter Weidong's password, and then you're in.[13]

---

[10] Talk to Julian Monroe or Sam Maxie on the 6th floor if you want to sign up for a ugastro account. You'll get a username and a password which you can change. With it, you get access to IDL, a place to store your own files, and can login to hercules remotely.

[11] You would need to install a Unix-like environment such as Red Hat Linux or Exceed on your computer in addition to Windows or Mac. These are necessary if you want to be able to use a Unix-like desktop environment from your home computer and use the software on the computers you are logged into. Through programs such as telnet, ftp, and ssh, you can access your files remotely, but without Linux or a Unix emulator, if you just have Windows or Mac, you won't be able to use all of the other software on the remote computer, especially if it requires displaying graphics.

[12] Weidong can give you the password to hercules if you don't have it already. If ever confronted by a blank screen at hercules, type "Openwin" at the command line to get back to the standard desktop environment.

[13] The same keystrokes should work from home if you're running ssh, Linux, or Exceed.

3.2. Unix Basics

Once, you're logged in, you will want to open a window called an "xterm" in which you can start doing things on the command line. At Hercules, you right click with your mouse, and select "xterm". From computers in 705, in an OpenWindows Desktop environment, [14] some windows named /usr/local/ will already be open, and you just need to click on them and type "xterm &" at the command line prompt.[15]

hercules:~>	# what the command line prompt looks like from hercules
/home/friedman/ %	# what the command line prompt looks like from my ugastro account
>	# Arbitrary command line prompt, which I'll use mostly from now on
> xterm &	# Type this, and an xterm window will pop up for you to use

At this point, it might be useful to talk about the basic Unix directory structure and eventually list some important example commands that allow you to move in and out of directories, create and edit files, create new folders, and do the same standard things you would do in Windows. Windows has nested folders with files in them and Unix does too, but instead of clicking on folders, in Unix, you specify exactly where to go on the command line. For example, let's say, at ugastro, I begin in the directory /home/friedman/, which is called the **root** directory. If I want to go to a sub folder of /friedman/ named Photometry, at the command line, I type:

> cd Photometry	# to open a subdirectory, or I could also type
> cd /home/friedman/Photometry	# this is the full pathname to the directory
	# "cd" stands for "change directory"

The second line gives more information than I need, since /Photometry is an immediate subdirectory of the root directory, but let's say I was in the root directory and wanted to immediately jump to a subdirectory of /Photometry called /1998dh which has a further subdirectory called /B. In that case, from the root directory, I could not simply type:

> cd B	# Fails since B is not an immediate subdirectory
> cd /home/friedman/Photometry1998dh/B	# I would have to type the full pathname

That's the basics of Unix directory structure.

[14] In any case, there are various desktop environments supported by Unix, some of which have more user friendly icons to click on, but all rely primarily on the command line. The most common are OpenWindows and Common Desktop Environment (CDE). Hercules uses OpenWindows by default, and this is what I personally prefer. If you log in from 705, before you enter your username and password, you can click "Options" and then "Session" to select your preferred Unix desktop environment. You can even color code your windows, and create a background desktop theme, but you'll have to find out about that elsewhere.

[15] In general, I'll put the command line stuff on the left, and any comments about what the command does after the "#" symbol on the right. This is to make it clear that you don't type the comment into the command line.

Aside from just moving in and out of directories, there are a number of useful Unix commands to deal with directories, files, and whatnot. The major ones are summarized below.

3.2.1. Table of Useful Unix Commands

> ls	# lists the files and folders in the current directory (ls means "list source")
> pwd	# Names the directory you're currently in (pwd means "print working directory)
> cd	# changes to the root directory (cd stands for "change directory")
> cd ..	# changes to the directory one level up (cd ../.. goes 2 levels up and so on)
> cd *directory name*	# changes to the *subdirectory* you name.
> cd *full path name*	# changes to a directory from anywhere on your machine.
	# example: > cd /home/friedman/Photometry/1998dh
> mkdir *directory name*	# creates a new directory (mkdir stands for "make directory")
	# example: > mkdir OLD
> more *filename*	# lets you view the contents of a text file; scroll with spacebar and enter
> cp *file1 file2*	# copies file1 to file2 (cp stands for "copy")
	# example: > cp PhotometryPrimer.tex PhotometryPrimer2.tex
> mv *file1 file2*	# renames file1 to file2 (mv stands for "move")
	# example: > mv sn1998dhB0725.fit B0725.fit
> rm *filename*	# deletes the named file, (can also delete many at once) (rm means "remove")
	# example: > rm B0725.fit B0726.fit B0730.fit
> paste *file1 file2*	# pastes two text files together as adjacent columns
> cat *file1 file2*	# appends the first file with the second file (just adds more rows)
> lp *filename*	# print the named file
> xterm &	# opens an xterm and runs in the background
> clear	# clears the screen
> xlock	# locks the screen, must re-enter password to login again
> ls *.fit	# This is an example of command using a wildcard, which lists all the
	# files that end in ".fit". (The wildcard * can stand for anything)
> ls * > file1	# the ">" outputs the result of the first command to a file called file1
	# here file1 contains a list of all the filenames in the directory
> man	# Access the Unix help manual (usually, its much easier to just ask someobody)

Table 3: I exclude things like file permissions, command flags, the grep, and the | pipe commands.

Some other convenient Unix tricks are that you can go back and forth through the history of previous commands by typing the "up" arrow for previous commands, and the "down" arrow to go the other way. You can auto-complete text with unique beginning characters by hitting the 'Tab' key. You can also use the mouse to highlight filenames or directory names and paste the text directly into the command line by middle clicking. You can also enable an up down scrollbar by hitting "Crtl-middle click" with your mouse inside an xterm, and selecting "Enable Scrollbar".

3.2.2. *Text Editors:* ***vi*** *and* ***emacs***

To create and edit text files in Unix, you use text editors, just like Microsoft Word for Windows. The two standard text editors for Unix are **emacs**, and **vi**. Of the two, emacs is much more user friendly than vi, since the former is mouse based, and the latter is keyboard based, but both can be powerful. This is not a primer on text editors, so I'll just list a few useful basic commands in each. To open either of them, at the command line, type:[16]

> vi *filename*	# to either create or edit a file with that name
> emacs *filename* &	# to either create or edit a file with that name. & means "run the program
	# in the background". It allows you to run several programs from one xterm.

emacs	
Left Mouse Button	# Highlight Text
Middle Mouse Button	# Paste Text
Search: Query Replace	# Search and Replace
Ctrl-x-s	# Save current file

vi	
Esc	# Safe mode: Allows you to scroll with arrows, can't edit text
i	# insert text before cursor
a	# insert text after cursor
R	# replace character at cursor
x	# delete character at cursor
dd	# delete a single line
y6y	# copy 6 lines, or any other number
p	# First scroll to the desired spot, then hit 'p' to paste the lines you just cut
u	# undo
ZZ	# save and quit
:wq	# write and quit
:1,$ s/B/BB/g	# search from beginning to end, replace every "B" with "BB". Without the "g",
	# which means global, it will only replace the first occurrence in each line.
:1,$ s/∧/mv /	# search from beginning to end, replace beginning of line with "mv "

[16]vi runs in the current window and emacs opens a new window, so it is useful to run emacs in background.

3.2.3. Command Aliases

Another extremely useful thing is to define what are called command aliases. An alias is essentially a shortcut for a command you would type on the command line. So, for example, if I didn't want to have to type "cd /home/friedman/Photometry all the time", I could define an alias such that I only had to type something shorter, like "ph". To create an alias, you have to edit a file in the root directory called ".cshrc", which controls the setup parameters of your Unix environment. To edit the file, use either vi or emacs and just add a line in the form of:

alias ph 'cd /home/friedman/Photometry'

to the end of the file, but don't touch anything else or it could screw things up. Once you logout and then login again, the new alias should work. Here are some aliases that I find useful.[17]

alias c	'/usr/bin/tput clear; ls -C'
alias x	'xterm &'
alias n	'netscape &'
alias p	'/usr/bin/lp -d rrp705'
alias hercules	'slogin -l weidong hercules.berkeley.edu'
alias ph	'cd /home/friedman/Photometry;clear;ls -C'
alias 98df	'cd /ibm/weidong/sn1998df;clear;ls'-C

Table 4: Some Example Unix Aliases. The last two are examples of shortcuts to specific directories in my ugastro account, and at hercules. In general, make an alias for anything you type often.

Unix has other software you might find useful. Instead of clicking on the shortcut icon as in Windows, here you open a program by naming it on the command line. Here are some examples:

> netscape &	# Opens netscape, the default internet browser
> xv *filename* &	# Opens xv, a graphics creation device, can grab individual windows
> ghostview *filename* &	# Opens ghostivew, which can read .ps (postscript) files such as this primer
> acroread *filename* &	# Opens Adobe Acrobat Reader, to view .pdf (portable ducument format) files
	# You generally don't need to specify a filename to open the program

Table 5: Some Useful Unix Programs. The ampersand "&" after the command indicates that you want the command to run in the background. For practical purposes, this means that you can run another program at the same time. If you don't include the "&", then you won't be able to do anything more on the command line in that xterm until you close the program that you opened.

[17]In general, you must be careful not to make aliases that conflict with standard Unix commands like ls and cd, since aliases usually take precedence. And despite the exceptions in this list, it is generally better to make aliases that exceed a single character. A 5-6 character alias still beats typing the full pathname for a deeply nested directory.

4. Preparing Your Data Set: Get a Feel For the Data

You'll find that supernova data sets (and data sets in general) can vary widely in quality, so it's a good idea to look through the images in your data set and get a feel for the data before starting photometry. You want to look at your finding chart and identify the supernova and the bright standard stars, and ultimately display the images and get a sense for the image quality in all the bands. Each image represents a data point in your light curve. The higher the quality of the data points, and the more data points you have, closely spaced in time, the better your final light curves will be in coverage and accuracy. In this section, we'll learn how to start IRAF, about the different image file types, and how to organize and look at your data. Then we'll look at how to create lists of the image names and Julian dates, both of which will be useful to printout and keep around as references. Then we'll talk about how to display all the images and get a feel for the data set. These are some necessary preparations steps before beginning photometry.

4.1. Starting IRAF

Before you can display the images, you need to be in IRAF, so here's how to start it. Weidong will make an IRAF directory for you with a name like /iraf_andy or /iraf_brandon, for example. In that directory will be a file named "startiraf" with instructions to start IRAF. This is just a command line script that effectively types in several things at the command line for you. If you've done the supernova search, you've already learned to first open and ximtool display window by right clicking the desktop at Hercules, and seen the command sequence to start IRAF:

```
>cd iraf      # > cd irafm also works, (Maryam Modjaz' old IRAF)
>cl           # enable the "command line" cl, in IRAF
>setimt8      # if you use ximtool8, for example.
```

In order to make sure people don't use the same ximtool simultaneously, each person doing photometry will get their own one. This will be built into your individual startiraf script, so you don't even have to remember to type "ximtool8 &", for example. startiraf does everything you need, so instead, all you do is go to your IRAF directory and type:

```
>startiraf
```

Now you'll be ready to go and the command line prompt will look like[18]

```
cl>
```

[18]In general, if you need to already be in IRAF to run some command, I'll use cl> for the command line prompt, and > for commands that will work generally. But its also true that some normal commands no longer work in IRAF or in IDL. For example, to make aliases work in IRAF, you need to first type ! before the alias. In general, all Unix commands are preceded by "$" or "spawn, '*command*' " in IDL.

Another thing that will be stored in your personal IRAF directory will be a list of all the parameters you set when you do photometry. These are stored in a file named .uparm. We'll talk about what the parameters are later, but suffice it to say, you want to make sure you use the parameters you set, rather than those used by someone else doing photometry. This is why you do everything from your own IRAF directory. Everyone has their own parameters which will not be overwritten by anyone else as long as they start IRAF from their own directory.

4.2. Image File Types

The types of image files you will be dealing with are generally of the types: .fit, .imh, and .fts. These are simply alternative image file types, comparable to the standard ones you might be more familiar with like .jpeg or .gif files. The data you get from Weidong will be in some directory he will specify, for example /ibm/weidong/sn1998dh/andy or /hale/weidong/sn1998ef. [19] The images will initially be .fts files, and they may be compressed to save space. In that case, the filenames will have an additional .gz or a .zip extension and look like .fts.gz or .fts.zip or .fts.z. In that scenario, you have to uncompress or unzip them to acces the images. Once you're in the directory with the data, this can be done with the Unix commands

```
>uncompress *.z    # uncompresses all compressed files in that directory with a .z extension
>gunzip *.gz       # unzips all zipped files in that directory with a .gz extension
```

You'll eventually convert .fts files to .fit files, and you might generate .imh files if you eventually do galaxy subtraction, which will be discussed later. If you've done the supernova search, you've already seen .imh files.

Here are just a few useful things to know about .fit and .imh files. .fit and .imh files have a header that contains important information about the file. You can access them in IRAF with the imhead and grep commands. Here are a few of the most important ones.

cl> imhead *imagename*.fit | grep FILTER
  # outputs the filter in which the image was taken, (i.e, "B")
cl> imhead *imagename*.fit | grep UT
  # outputs the Universal Time (UT) at which the image was taken (i.e 6:41:34)
cl> imhead *imagename*.fit | grep DATE-OBS
  # outputs the date of observation of the image in the form (dd/mm/year) (i.e 20/10/1998)

If you don't use the | and grep commands, imhead will output the entire image header, which has a lot of useless information. The same commands work for .imh files as well, as they also have an image header that can be read by IRAF.

[19] hercules has 4 hard drives, ibm, whale, hale, and vela, so it will be standard to store and access your data on one of these that has space left. Photometry data can take up a lot of space, so its good to keep disk space in mind, and not make superfluous extra copies of files, for example.

A useful thing to know about .imh files is that the image header and the actual pixel information (a .pix file) are stored in separate places on the hard drive. .imh itself stands for image header. So if you have to delete an image with a .imh extension while in IRAF, use

cl> imdelete *imagename*.imh   # Do not use the "rm" command, since it will only delete the header,
and not the pixel information, which takes up a lot of disk space.

4.3. Organizing Your Data

Now that you have an idea about your data files, the next thing to do is to organize the data into directories for each of the different filters. In the past, you had to do these things manually:

- Unzip all the compressed images and change all the .fts files to .fit files
- Use imhead to find out the filters for all the images
- Use crazy vi tricks to rename all of your image files to nice standard names such as B07251998.fit, (standing for the B band image taken on 7/25/98)
- Finally you had to manually put them into the directories U,B,V,R and I (as needed).

Fortunately, Weidong has written a program called "classfile", which does all of this for you. So now, all you need to do is go to your data directory and type

>classfile

And it will automatically uncompress the files, find the filter in the headers, rename the file, and sort them into the relevant directories. If, for example, you don't actually have any U band data, it will still create the folder U, but it will simply be empty, so you can delete it.

4.4. Creating and Printing a List of Images

Before displaying the images, one other useful thing to do is to create and print out a list of image names. You can use this as a reference and can take notes on it throughout the process. For example, it is useful to remember how many images are in each band, which days were observed in all bands, and which days have gaps in some bands. It helps to create a file where the image names for each band are in columns, and the individual rows correspond to an individual day of observation. If days are missing in some band, leave that space in the row blank. In any case, here's how to make it. You don't need to be in IRAF. First, go to the B directory, for example. Then type:

> ls *.fit > blist   # outputs a list of all files ending in .fit to a file named blist (* = wildcard)

now go to the other directories U,V,R, and I, and repeat this process, creating the files ulist, vlist, rlist, and ilist, or whatever you choose to call them. Now, move all these files to any directory, for example, to the main data directory. In the B directory, for example, you can type:

>mv blist .. # moves the file blist to the directory one level up, the main data directory

Do the same for each of the other files so they're all in the same directory. Once they're in that directory, go there and type:[20]

>paste blist vlist rlist ilist > bvri_list # pastes all files together as adjacent columns

Now if you edit the file with

>vi bvri_list # you'll see a file that might look like the leftmost list on the next page

If you look at the leftmost list on the next page, you'll notice that there are a different number of images in each band: B (40), V (41), R (40), and I (41). Upon closer inspection, we see that the B image from 7/27 and the R image from 8/28 are missing. Thus, we want to edit the file to align the filenames such that each row corresponds to a unique day of observation. Thus we use vi or emacs to make the file look like the rightmost list on the next page.

You can print this list with the lp command

>lp bvri_list # at Hercules
>lp d rrp705 bvri_list # at 705, -d is some flag, and you have to name the printer, rrp705

Once printed, this sheet is a useful reference to keep around. You may even want to print out several copies of this list for various note taking purposes. You'll need one for keeping track of the good and bad images you display, and eventually for keeping a record of what you did for dark current subtraction and for cosmic ray subtraction. They really do come in handy. One other thing to note. Although it is not the case in the example lists on the previous page, the final images in the list might be galaxy templates, and thus, not a standard part of the supernova data set. Keep track of this fact in the future, and either separate them from the other entries in the printed list, or simply delete them from bvri_list.

[20](Assume there's no U band data for now) Also, if you're in IRAF, paste becomes !paste

B0725.fit	V0725.fit	R0725.fit	I0725.fit
B0726.fit	V0726.fit	R0726.fit	I0726.fit
B0728.fit	V0727.fit	R0727.fit	I0727.fit
B0729.fit	V0728.fit	R0728.fit	I0728.fit
B0730.fit	V0729.fit	R0729.fit	I0729.fit
B0731.fit	V0730.fit	R0730.fit	I0730.fit
B0801.fit	V0731.fit	R0731.fit	I0731.fit
B0802.fit	V0801.fit	R0801.fit	I0801.fit
B0803.fit	V0802.fit	R0802.fit	I0802.fit
B0804.fit	V0803.fit	R0803.fit	I0803.fit
B0806.fit	V0804.fit	R0804.fit	I0804.fit
B0807.fit	V0806.fit	R0806.fit	I0806.fit
B0812.fit	V0807.fit	R0807.fit	I0807.fit
B0813.fit	V0812.fit	R0812.fit	I0812.fit
B0815.fit	V0813.fit	R0813.fit	I0813.fit
B0816.fit	V0815.fit	R0815.fit	I0815.fit
B0817.fit	V0816.fit	R0816.fit	I0816.fit
B0818.fit	V0817.fit	R0817.fit	I0817.fit
B0820.fit	V0818.fit	R0818.fit	I0818.fit
B0822.fit	V0820.fit	R0820.fit	I0820.fit
B0824.fit	V0822.fit	R0822.fit	I0822.fit
B0826.fit	V0824.fit	R0824.fit	I0824.fit
B0828.fit	V0826.fit	R0826.fit	I0826.fit
B0830.fit	V0828.fit	R0830.fit	I0828.fit
B0901.fit	V0830.fit	R0901.fit	I0830.fit
B0903.fit	V0901.fit	R0903.fit	I0901.fit
B0913.fit	V0903.fit	R0913.fit	I0903.fit
B0917.fit	V0913.fit	R0917.fit	I0913.fit
B0919.fit	V0917.fit	R0919.fit	I0917.fit
B0920.fit	V0919.fit	R0920.fit	I0919.fit
B0921.fit	V0920.fit	R0921.fit	I0920.fit
B0925.fit	V0921.fit	R0925.fit	I0921.fit
B1009.fit	V0925.fit	R1009.fit	I0925.fit
B1013.fit	V1009.fit	R1013.fit	I1009.fit
B1017.fit	V1013.fit	R1017.fit	I1013.fit
B1021.fit	V1017.fit	R1021.fit	I1017.fit
B1114.fit	V1021.fit	R1114.fit	I1021.fit
B1126.fit	V1114.fit	R1126.fit	I1114.fit
B1217.fit	V1126.fit	R1217.fit	I1126.fit
B1218.fit	V1217.fit	R1218.fit	I1217.fit
	V1218.fit		I1218.fit

B0725.fit	V0725.fit	R0725.fit	I0725.fit
B0726.fit	V0726.fit	R0726.fit	I0726.fit
	V0727.fit	R0727.fit	I0727.fit
B0728.fit	V0728.fit	R0728.fit	I0728.fit
B0729.fit	V0729.fit	R0729.fit	I0729.fit
B0730.fit	V0730.fit	R0730.fit	I0730.fit
B0731.fit	V0731.fit	R0731.fit	I0731.fit
B0801.fit	V0801.fit	R0801.fit	I0801.fit
B0802.fit	V0802.fit	R0802.fit	I0802.fit
B0803.fit	V0803.fit	R0803.fit	I0803.fit
B0804.fit	V0804.fit	R0804.fit	I0804.fit
B0806.fit	V0806.fit	R0806.fit	I0806.fit
B0807.fit	V0807.fit	R0807.fit	I0807.fit
B0812.fit	V0812.fit	R0812.fit	I0812.fit
B0813.fit	V0813.fit	R0813.fit	I0813.fit
B0815.fit	V0815.fit	R0815.fit	I0815.fit
B0816.fit	V0816.fit	R0816.fit	I0816.fit
B0817.fit	V0817.fit	R0817.fit	I0817.fit
B0818.fit	V0818.fit	R0818.fit	I0818.fit
B0820.fit	V0820.fit	R0820.fit	I0820.fit
B0822.fit	V0822.fit	R0822.fit	I0822.fit
B0824.fit	V0824.fit	R0824.fit	I0824.fit
B0826.fit	V0826.fit	R0826.fit	I0826.fit
B0828.fit	V0828.fit		I0828.fit
B0830.fit	V0830.fit	R0830.fit	I0830.fit
B0901.fit	V0901.fit	R0901.fit	I0901.fit
B0903.fit	V0903.fit	R0903.fit	I0903.fit
B0913.fit	V0913.fit	R0913.fit	I0913.fit
B0917.fit	V0917.fit	R0917.fit	I0917.fit
B0919.fit	V0919.fit	R0919.fit	I0919.fit
B0920.fit	V0920.fit	R0920.fit	I0920.fit
B0921.fit	V0921.fit	R0921.fit	I0921.fit
B0925.fit	V0925.fit	R0925.fit	I0925.fit
B1009.fit	V1009.fit	R1009.fit	I1009.fit
B1013.fit	V1013.fit	R1013.fit	I1013.fit
B1017.fit	V1017.fit	R1017.fit	I1017.fit
B1021.fit	V1021.fit	R1021.fit	I1021.fit
B1114.fit	V1114.fit	R1114.fit	I1114.fit
B1126.fit	V1126.fit	R1126.fit	I1126.fit
B1217.fit	V1217.fit	R1217.fit	I1217.fit
B1218.fit	V1218.fit	R1218.fit	I1218.fit

Table 6: On left: bvri_list (before editing): columns have varied lengths, rows don't always correspond to the same day of observation. On right: bvri_list (after editing): columns have same length, rows correspond to same day of observation. Missing images in a band are left blank.

4.5. Creating a List of Julian Dates for Each Band

In looking at the edited version of bvri_list, you may have noticed that since each row now corresponds to a different day of observation, each column essentially represents a time axis. Eventually, you will want to create a time axis for each band measured in units of Julian days, so you might as well do itnow. The time axes for each band will be similar because usually the B band and V band observations might only be taken a few minutes apart, and this is only a small fraction of a day. Nevertheless, it becomes useful to create a list of Julian dates for each band. Eventually each of these lists becomes the x-axis of your light curve, as you plot your photometric data (in magnitudes) vs. time (in Julian days). So how do we create a list of Julian dates?

First of all, the KAIT server has a convenient program called jd, which outputs the Julian date if you tell it the calendar date and time of day (in UT) when the observation was taken. From Hercules, simply type"

```
>kait      # to log into the kait server remotely, you've may have done this before
           # to create request files for the supernova search
```

Once there, the prompt will look like

```
kait%
```

To run the Julian date program, here is an example call sequence

```
kait% jd ut=06:41:34 date=20/10/1998
```

after which the program outputs the relevant Julian date in decimal form

```
kait% 2451106.7789
```

So now you want to do this for all of the images in all of the bands, but first, you have to get the Universal Time (UT) and Date of Observation (DATE-OBS) from the image header. Then you have to create a list of command for each of the images, and run those commands on kait. Here's how to do it, for the B band, for example. You must be in IRAF, and in your B directory.

```
cl> imhead *.fit | grep UT > b_UT   # outputs a list of the UT's for all the .fit images to the file
                                    # b_UT. If some .fit files at the end of the list are galaxy
                                    # templates, you can remove them from the list later


cl> imhead *.fit | grep DATE-OBS > b_DATEOBS     # outputs a list of all the DATE-OBS for all the
                                                 # .fit files to a file named b_DATEOBS
cl> !paste b_UT b_DATEOBS > b_jdscript           # in IRAF paste becomes !paste
cl> vi b_jdscript                                # Now view and edit b_jdscript
```

```
# the first few lines of the file should look something like:
UT      = '06:41:34'     DATE-OBS = '20/10/1998'
UT      = '07:00:07'     DATE-OBS = '21/10/1998'
UT      = '06:16:34'     DATE-OBS = '22/10/1998'
UT      = '06:15:19'     DATE-OBS = '23/10/1998'
```

now edit the file, for example in vi, by doing these kinds of search and replace commands:

```
:1,$  s/UT = '/jd ut=/
:1,$  s/DATE-OBS= '/date= /
:1,$  s/'//                    # repeat 3 times
Esc                            # get out of editing mode
ZZ                             # save the file and quit vi
```

You could easily use the Query Replace feature of emacs as well. In the end, the first few lines of the file named b_jdscript should now look something like this, exactly the required call sequences for jd:

```
jd ut=06:41:34 date=20/10/1998
jd ut=07:00:07 date=21/10/1998
jd ut=06:16:34 date=22/10/1998
jd ut=06:15:19 date=23/10/1998
```

Now to run them on the kait server, you could transfer the file over there using ssh or ftp and add "#!/bin/csh" to the first line to make it executable, but there's a simpler way. First open a separate window and log into kait. Then just open b_jdscript with emacs, in your original window, copy the the text with your mouse and paste it directly onto the command line in kait and hit enter. It should output a list of the Julian dates in order on the kait server. Now highlight these with your mouse, open a new emacs file, and paste them *in order* into the new emacs file. Make sure to get all of them, and get rid of the "kait%" characters that you might copy. Save it and name it something like b_jd. Now repeat this process for each directory and you'll have a list of Julian dates in each band. For example, in B, the first few lines of b_jd should look something like:

```
2451019.9312
2451020.9366
2451021.0000
2451022.9468
```

Alternatively, since 1 Julian day = 1 UT day, you could just get the Julian day for the first image in each band using the jd program on KAIT, and then just recognize that since you already have all the UT days in a list, you can deduce the remaining Julian days just by offsetting by some number of decimal days. But this method would probably require you to write a small program, so for now, you can just use the method I described. But if you want to do it this way, which can be simpler in principle, just ask Weidong for help in writing a small program.

Using the list of Julian dates in each band, you can now create a file with columns alternating between Julian dates and image names for each band, using the same kinds of techniques of pasting files together from before. For example, if you have the files for each band with names like blist and b_jd, you could paste them together like so

> paste b_jd blist v_jd vlist r_jd rlist i_jd ilist > bvri_jd_list

Now you can use the file bvri_list to note which days are missing and makes sure everything in each row corresponds to a unique date of observation, although the Julian dates themselves will differ in their decimal places since the images were taken a few minutes apart in each of the bands.

Aligning the images along the time axis can be extremely important later because you may be writing programs to plot the data that involve performing operations on arrays of data, with the assumption that, in all the arrays, the same array index will correspond to the same date of observation. If the arrays are displaced in time by even a single day, it could skew your light curve noticeably. Early in the light curve, being off by even a single array index, and probably a single day might matter a lot since the supernova changes in brightness faster in early times. In late times, the supernova changes in brightness slower, but being off by one image date will introduce a magnitude error since the observations are spaced out further in time because the late time light curve is generally not as important yet for cosmology, and need not be as well sampled. In either case, it is crucial to have your time axes aligned from the start.[21]

4.6. The Finding Chart

The finding chart is just a reference printout of a good image of your field, which has the supernova and a bunch of other stars in the field called "secondary standard stars" or "field standards", where all the stars are labeled with numbers.[22] An example finding chart can be found on the next page. In the finding chart, the supernova is generally numbered "1", with the field standards following "2,3,4..." in some arbitrary order. Although, once you number the field standards, you must keep track and use this numbering system throughout the entire photometry process. In addition to the secondary or field standards, the primary standard stars come from a separate catalog, such as the Landolt catalog, and are generally not in your field (unless you're lucky). Eventually you use the primary and secondary standards together to calibrate your supernova light curve.

---

[21] A shift in the time axis of the arrays won't affect your instrumental light curve, if you plot each band vs. its own time array of Julian dates, but for the standard light curve, it will matter since the equations to transform from instrumental to standard magnitudes involve cross terms between bands, which may have slightly different individual time arrays, for example, if an image is missing in the B band but not from the V band.

[22] Weidong can give you a finding chart, or, as discussed later, you can create one yourself once you learn how to make a coordinate file and use the tvmark command.

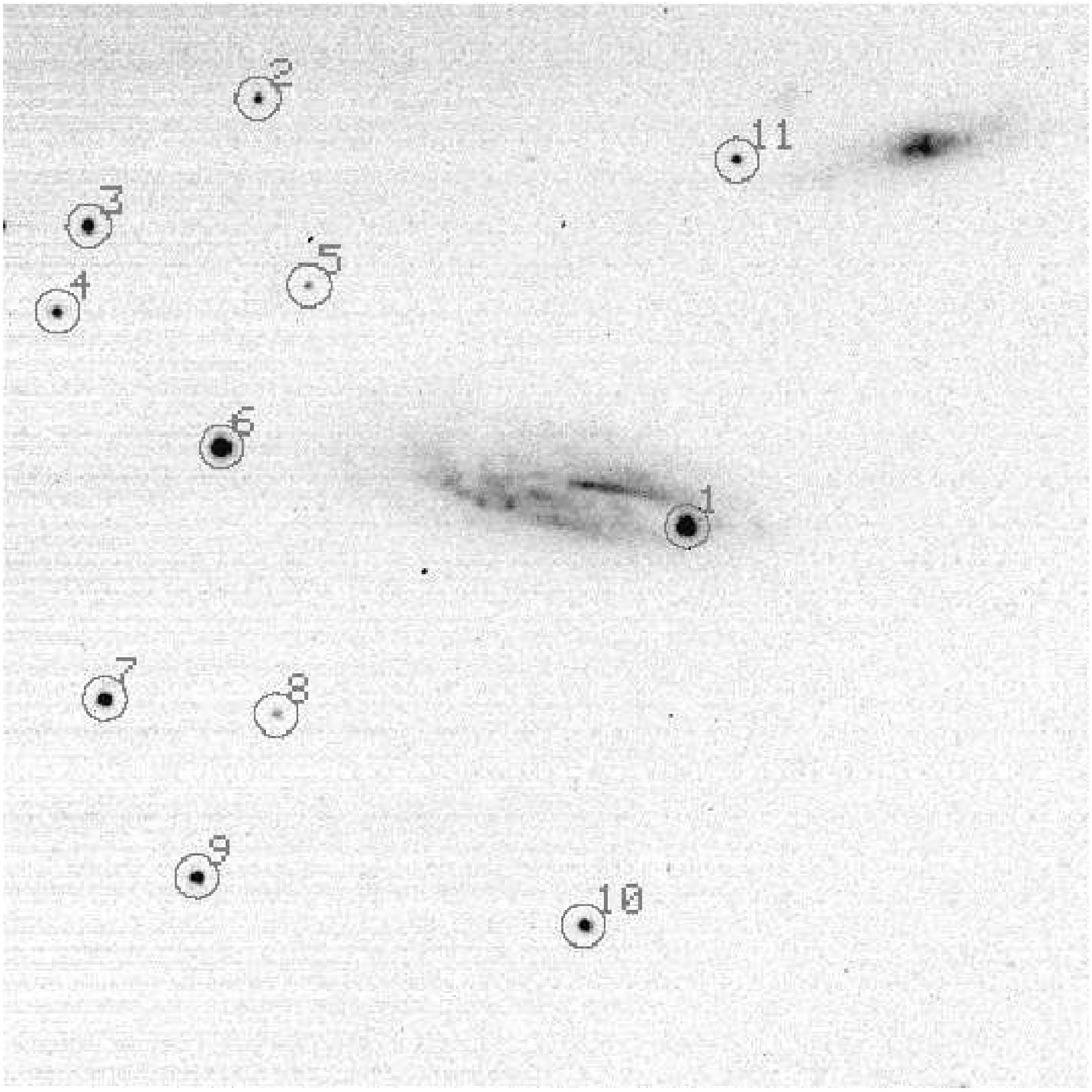

Fig. 7.— An example finding chart for supernova 1998dh. In the chosen image, which happened to be taken on 8/2/98, the supernova is numbered "1" and the secondary standard stars in the field are numbered "2,3,4..." in an arbitrary order which you choose at the beginning, but which you must keep consistently throughout the process. The finding chart is a useful reference to keep at hand while doing photometry.

4.7. Displaying the Images

Now finally, we're ready to display the images. If you've done the supernova search, you've already seen the display[23] command which just so happens to display an individual image in IRAF where an ximtool is already open.

```
cl> display imagename 1     # displays the named image in frame 1 of the ximtool
```

You can get other useful information from the imexam command

```
cl> imexam imagename 1   # a cursor will pop up in the ximtool, and you can examine the image
                         # in a bunch of ways with various keystrokes. Here are a few of them
                         # that you may already be familiar with from the supernova search
s                        # displays a surface plot in a separate Tek window
e                        # displays a contour plot in a separate Tek window
r                        # displays a radial plot in a separate Tek window
a                        # hit 'a' on a star (or SN) and it gives info such as the FWHM
                         # (Full Width Half Maximum) of the star which is called "GAUSSIAN"
                         # Overall 'a' gives: COL, LINE, COORDINATES,
                         # R, MAG, FLUX, SKY, PEAK, E, PA, ENCLOSED, GAUSSIAN, DIRECT
m                        # or if you hit 'm' on empty space you can get the number of background counts
                         # called "MEAN" or the standard deviation of the sky background,"STDEV".
                         # Overall,'m' gives: SECTION, NPIX, MEAN, MEDIAN, STDDEV, MIN, MAX
q                        # quit
```

But instead of displaying all the images manually, we can use a vi trick to display them all in order for each band, and you won't have to type in the display command for each one. In your B directory, for example, type:

```
>ls B*.fit > blist       # you may have already done this when creating the list of Julian dates
                         # make sure to remove or note the galaxy template names in this list
>cp blist b_display      # copies blist to a file named b_display
>vi b_display            # edit b_display in vi

:1,$  s/^/imexam /       # search for the first blank space in each line, and replace it with "imexam "
Esc
ZZ
```

[23] You could also type "cl> disp imagename 1" in IRAF. If the "disp" part is unique, Unix and IRAF will recognize the full command. You could also type "imdel" instead of "imdelete" and so on.

Now, if you're not already in, you must go to your IRAF directory, and run startiraf, which will open your ximtool. Now go back to your data directory where you've just created the file b_display and type:

cl>cl < b_display   # Typing "cl" makes the file b_display an executable IRAF script and runs it
The first image in the list will pop up in the ximtool in imexam mode.
Here you just want to look through them so hit 'q' to go to the next image

Here are some typical questions to keep in mind, and to take notes on, when looking through the images in each of the bands:

- Was there bad weather on some nights? Do stars look blurred or not starlike?
- Was there ever a very high level of background noise in the image? (Maybe the moon was bright?)
- Were there any errors in image processing? Is the supernova in the image?
- How good is the temporal coverage ? Are the observations numerous and well spaced in time?
- How early before maximum brightness did we catch the supernova?

If you can't see the supernova in the image, you can't do anything with it, and you'll have to throw it out. If the image quality is just really bad, and would lead to a photometric data point with extremely large error bars, you may need to justify throwing it out later. This is all stuff you can take notes on, writing anything you think is relevant on one of the printouts of image names you just made. This helps you to keep track of the bad images, and ones with weird features that you might throw out later, it keeps things organized, and it saves you a lot of time.

5. Cleaning Your Data Set: Dark Current and Cosmic Ray Removal

Finally, we'll talk about how to clean the images to remove dark current and cosmic rays before you can begin photometry. If not removed, these effects can introduce possible systematic errors into our magnitude measurements. In the end, in an image, we want to count up only the light, or the number of photons that hit our CCD, that came from the supernova and standard stars. Dark current and cosmic rays are additional sources of "counts", which are not due to light from the objects we are interested in, thus they must be removed.[24]

5.1. What is Dark Current?

To explain what dark current is, we must first discuss a bit about how the CCD camera works. CCD itself stands for Charge Coupled Device, and it is the standard technology used for the light detector chips in astronomical instruments, and more recently, in modern digital cameras.

[24] Later on, when we talk about galaxy subtraction, we will be doing something similar, where we remove the counts that are due to light from the host galaxy, leaving only the light from the supernova, which we are interested in.

It is generally a square pixel array (say 500 x 500 or 256 x 256) where each pixel, often made of semiconductor silicon, detects photons via Einstein's photo-electric effect. At the basic level, incoming photons knock electrons off the detector elements (pixels), creating a photo-current of electrons that can be traced back to individual pixels and used to reconstruct the image. By this process, each pixel registers a certain number of photon counts over the exposure time of the image.

When you take an image in a given wavelength band, the CCD camera essentially "counts" the number of transmitted photons that hit each pixel during the exposure, but of course, does so imperfectly. Part of the imperfection occurs when the CCD pixels fail to detect some of the incoming photons. The accuracy with which the CCD detects incoming photons is called its "quantum efficiency", which, as one would expect, varies as a function of the transmitted wavelength. So, in some wavelength band, if an average pixel has 100 photons hit it during an exposure but if only only 60 electrons are knocked off and measured as photo-current from that pixel, then we say the CCD has a quantum efficiency of about 60% in that band. Another source of imperfection, comes not when the CCD misses a few counts, but when it registers "false" counts that were not due to incoming light at all. The major source of these false counts is called dark current.

Dark current is the phenomenon where the detector will register "dark" or "false" counts as if a photon has been detected, even when the shutter is closed and the CCD is not receiving any external photons. This process actually has a fairly well understood thermodynamic and quantum mechanical basis. At a finite detector temperature, there is always some probability that an individual detector element or pixel (usually made of semiconductor silicon wafer), will emit an electron, thereby registering a false count. Since the probability of such random events decreases at lower temperatures, the effect of dark current can be drastically reduced simply by cryogenically cooling the CCD. The KAIT CCD camera is indeed cryogenically cooled, but it is impossible to completely prevent dark current. As a result, the effects of dark current are quite prevalent, and must be taken into account and removed after the fact.

5.2. Dark Current Removal

When you subtract the dark current for an image, ideally, you would have taken a corresponding dark image immediately afterwards with the same exposure time as the original image, and under nearly identical conditions. You should use the same exposure time and take the dark image immediately afterwards, because you want the dark images to have roughly the same number and distribution of dark counts as your science image, so you can subtract them. This requires nearly identical observational conditions where the deterctor temperature has not had much time to change. In this scenario, one would simply take subtract the dark image from the original image, pixel by pixel, count for count, and on average, this would remove much of the dark current. Unfortunately, taking a unique dark image for every observation is not the best use of observational resources for KAIT, which takes a lot of images, and whose efforts are best spent imaging galaxies and supernovae we've actually discovered. Thus, since we do not have corresponding dark images for every science image, we must approximate by taking a standard dark image for some exposure time, and using it to remove the dark current. An example of a standard dark image can be seen below.[25]

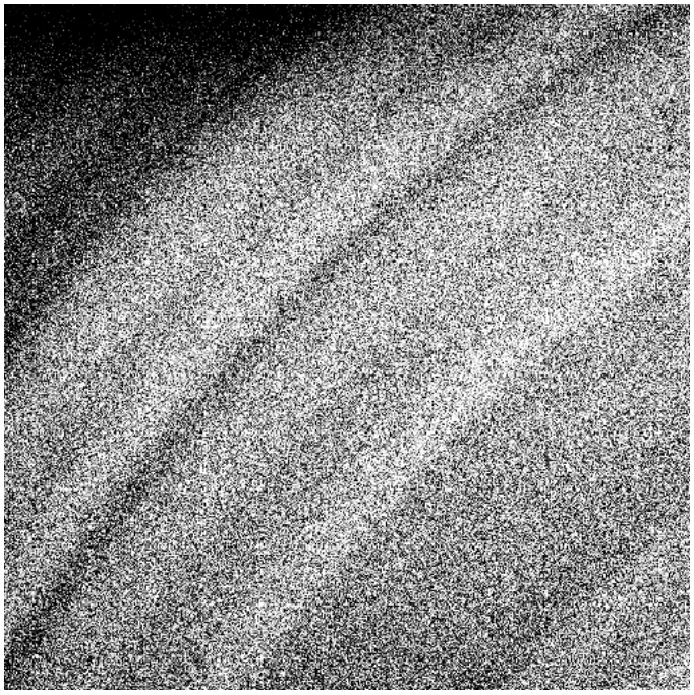

Fig. 8.— Example of a dark image, taken in 1998. You can see distinct bands running diagonally across the image. These correspond to the shape of the temperature gradient across the CCD when the image was taken with the shutter closed.

To remove the dark current from an individual supernova image, we thus must subtract (or add) some fraction of the dark image from our supernova image. This is because the two images may not have had the same detector temperature or exposure time, and the longer the dark exposure, and the hotter the camera was, the more dark counts there will be in each pixel on average. First, you have to decide which images need to have dark current removed. **In general, dark current subtraction is needed most in the U and B bands**. But occasionally, you will need to remove it from a few images in the other filters. Ideally, when you look through all the images, you can note the images where the dark current bands were noticeable, as in the above example image, and mark on the printout that these images will need dark subtraction.

To do so, in IRAF, you use the imarith command. Let's assume that the relevant standard dark image is named dark.fit, (Weidong will put this image in your directory at your request) and that the image you are removing the dark current for is displayed in your ximtool. Then you type:

[25]Every time we get a new CCD camera, or replace the optics in the telescope, we must make sure we have new dark images that that representative for the new CCD hardware and optics during their time of operation.

```
cl> imarith dark.fit * 0.1 d01                    # Creates an image named d02 where the pixel
                                                  # counts have 10% of the value of dark.fit
cl> imarith dark.fit * 0.2 d02                    # d02 is now 20% of dark.fit, and so on

cl> imarith imagename − d01 new imagename         # Subtract d01 from the image and output to
                                                  # a new file. For Example
cl> imarith B0725.fit − d01 B0725d.fit            # B0725d.fit = B0725.fit − d01
                                                  #"d" might stand for dark subtracted
cl> disp B0725d.fit                               # Display the dark subtracted image
```

Now you want to look at the dark subtracted image to see if the dark current bands are mostly gone, to determine whether you have successfully removed the dark current. If the dark current is essentially no longer visible, you can move the original image to a separate directory and rename the dark subtracted image back to the original image name, as long as you keep track of which ones have been dark subtracted. You could theoretically delete the original image, but its good to just put it into a separate directory temporarily just in case you mess up, and need to start dark subtraction over for that image. You can always delete the separate directory at the end, because .fit files do take up disk space. But you certainly should keep the original image around for the time being because the trial and error dark current subtraction might not work the first time.

If the dark current bands are still visible in the new image, start again with the original image and try some new combination. Maybe subtract d02, or add d01 or add d02. You certainly are not limited from creating images that, are say, 15% of dark.fit if you really want to fine tune it. But in general, this is an approximate method to remove the effects of dark current, and it's OK if you can't completely remove the dark current. Just play around with trial and error and choose the dark subtracted attempt that looks the best. Do this for all the images that need dark current subtraction, and you're ready to move on to the next step. This process does take some time, but, by inspection, you'll find that, in general, the U and B band images are the ones that require the bulk of dark subtraction.[26]

5.3. What are Cosmic Rays?

Cosmic rays are high energy particles from space that penetrate the Earth's atmosphere and can interact with the detector elements in your CCD camera, generating counts that are not due to light from the astronomical source you are imaging. 80% of the cosmic ray flux at sea level consists of muons, the higher mass cousin of the electron with the same electric charge. Other cosmic rays include protons, electrons, or even gamma ray photons, but for our purposes, we can just say that unspecified high energy particles from space can potentially hit the CCD camera and cause

[26] You do need to do dark subtraction on galaxy template images, if you have them, but again, it will probably only be necessary in the U and B bands.

problems, especially if a cosmic ray happens to hit a pixel on or near the supernova or standard stars. They are very high energy and can generate many photoelectrons, so in order to make sure these counts are not counted, we use IRAF software to remove cosmic rays from our image.

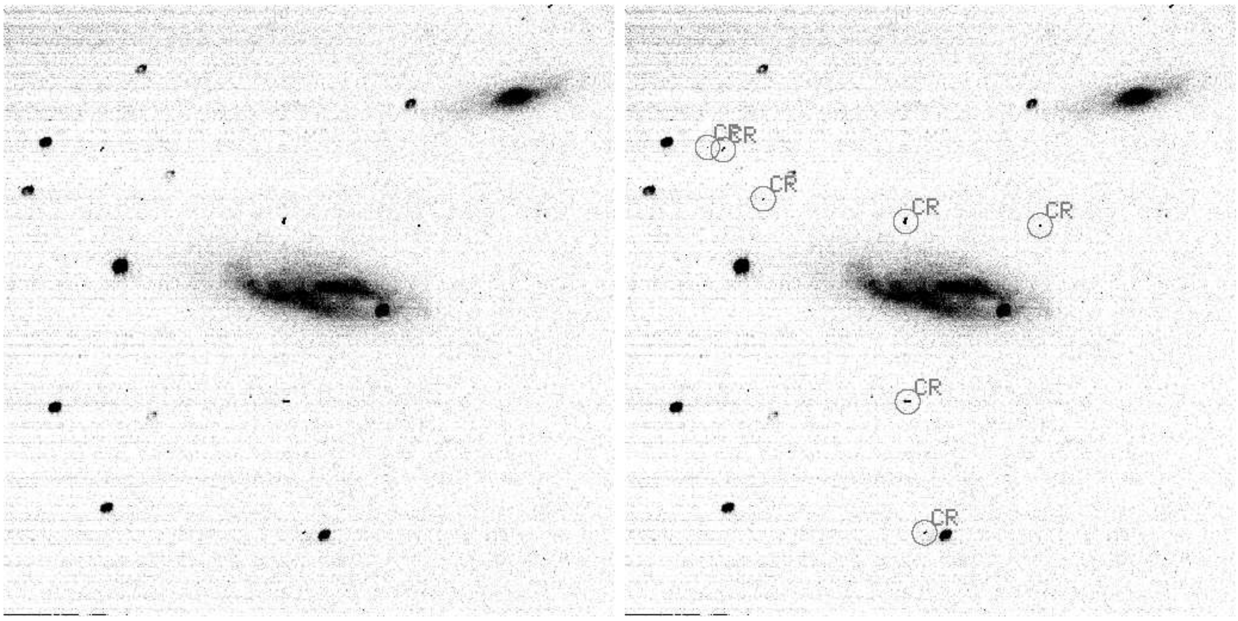

Fig. 9.— Here is an example of what cosmic rays look like in a B band image of supernova 1998dh taken on 8/20/98. As usual, there are number of cosmic rays in the image, which are highlighted with circles in the second frame.

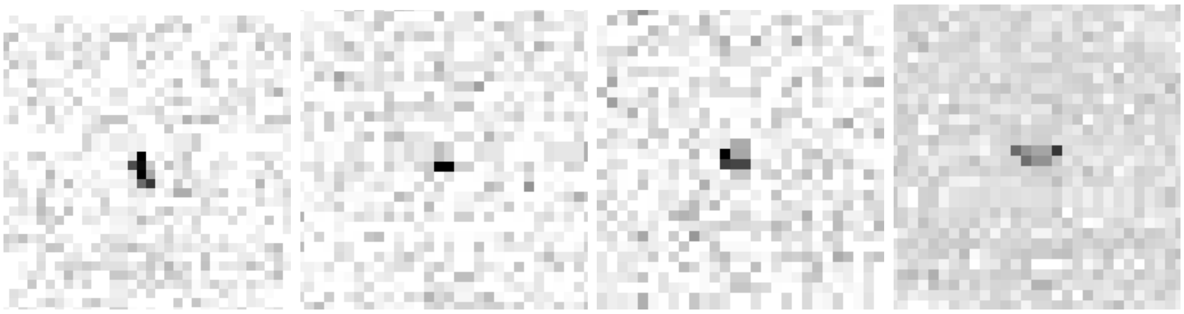

Fig. 10.— Here are a few examples of what individual cosmic rays look like on the CCD when this same image is magnified. Cosmic rays generally take up only a few pixels and sometimes just a single pixel. You are probably already familiar with this from the supernova search, where cosmic rays often appear, for a moment, to be good supernova candidates until you see that their profile is more like a spike or delta function than a starlike Gaussian.

5.4. Cosmic Ray Removal

Unlike dark current removal, all images require cosmic ray subtraction because during the exposure time of the KAIT image, there is a very high probability that a number of cosmic rays will strike the CCD. And in fact, it is very important to do dark current subtraction first, before you do cosmic ray subtraction, because the dark.fit image is typically a long exposure, which necessarily means it has a bunch of cosmic rays in it. If you reversed the order and did cosmic ray subtraction first, you would just be putting cosmic rays back in when you do dark subtraction. Thus it is crucial to do dark subtraction before cosmic ray subtraction.

You can remove cosmic rays manually or automatically, using the IRAF task **cosmicrays**. Since this is the first IRAF task you have seen so far (you'll see many more when actually doing photometry), the basic idea is that you edit a bunch of parameters related to that task and then run the task. If you do it manually, you make individual parameter choices for some image, then you run the task on that image in a setting where you can continually edit some of the parameters to fine tune the cosmic ray removal until you get it right. If you run it automatically, you set the parameters in advance, create a list of images to input, and then run the task on all those images automatically. When this runs it is essentially instantaneous. So the first thing to do is to go band by band, and separate the images into two categories for cosmic ray subtraction: manual and automatic.

You do this by separating the images according to whether or not they have a high sky background, or a high mean number of counts in the image in the spaces between the stars and galaxies in the image. The sky itself, or more accurately, the earth's atmosphere reflects and emits photons, some of which will hit the CCD. The moon, when bright and near the field of view of your image, also reflects sunlight and contributes photons that might hit your CCD. These contribute to the background counts in your image that are not from astronomical sources like other stars or galaxies. Now if the background counts are too high, they could overwhelm astronomical sources and drown them out, thus making the signal difficult to extract from the noise. This is one of the reasons why we can not see many faint stars from the ground, and this is a problem for all signal detection in general. But for our purposes here, what we are trying to detect are cosmic rays, and what we are trying to do is remove them. In order for the IRAF software to easily detect cosmic rays, it helps when the mean of the sky background is low, allowing the cosmic ray to look like a strong signal, or spike, that outshines the sky background by having a larger number of counts. This is when you can trust the software and run cosmicrays automatically. If the mean is very high, the software might have trouble, and you will have to bring some of your own judgment to bear and run cosmicrays manually.

So how do we find out what the value of the sky background or "MEAN" of the image is? You can use the IRAF command, imstat. For example, in the B directory, in IRAF, type:

```
cl> imstat B*.fit      # Will output some statistics for all the images, including "MEAN"
                       #(also IMAGE, NPIX, STDEV, MIN, and MAX)
```

For example, some selected lines of output for imstat B*.fit might look like:

#	IMAGE	NPIX	MEAN	STDDEV	MIN	MAX
	B0725.fit	250000	10.12	7.566	0.	1181.
	B0726.fit	250000	9.282	8.414	0.	1222.
	B0728.fit	250000	12.1	24.55	0.	1938.
	B0729.fit	250000	92.15	86.27	15.2	6391.
	B0806.fit	250000	1283.	49.38	360.5	7483.
	B0807.fit	250000	2018.	53.55	1369.	8750.
	B0812.fit	250000	1085.	44.16	538.2	7708.

In general, for images with a MEAN > 800, you need to run cosmicrays manually. If you have MEAN < 800, you can run cosmicrays and subtract the cosmic rays automatically. It might be helpful to print out a list and circle the image names which have MEAN > 800, where cosmic ray subtraction should be done manually.

```
cl> imstat B*.fit > Bstats.tab      # output the statistics to a file Bstats.tab
cl> lp Bstats.tab                   # and print it out (at Hercules)
```

Once you know which ones to automatically, you need to create a file with a list of the relevant filenames. To do this, type:

```
cl> ls B*.fit > Blist_auto     # output the .fit files to a file Blist_auto
cl> vi Blist_auto              # Now edit the file and delete the filenames
                               # of ones you are not doing automatically.
```

For example, from the above list, some entries of Blist_auto will be:

B0725.fit
B0726.fit
B0728.fit
B0729.fit

Conversely, the images

B0806.fit
B0807.fit
B0812.fit

all had MEAN > 800 and must be done manually. So before doing cosmic ray subtraction for any images automatically, we must first learn how to do it manually.

Now we can finally discuss how to use the IRAF task cosmicrays to perform cosmic ray subtraction. We will start by discussing how to run the task manually. First off, display the relevant image in your ximtool, then in IRAF, in the directory with the images you are working on, type:

cl> epar cosmicrays # edit the parameters (epar) of the IRAF task cosmicrays

Now a menu of parameters will pop up that looks something like the list below. The most important parameters for our purposes are highlighted here in bold.

I R A F
Image Reduction and Analysis Facility

PACKAGE	=	ccdred	
TASK	=	cosmicrays	
input	=		s1117.fit List of images in which to detect cosmic rays
output	=		t1.fit List of cosmic ray replaced output images (optional)
(badpix	=	)	List of bad pixel files (optional)
(ccdtype	=	)	CCD image type to select (optional)
(thresho	=	120.)	Detection threshold above mean
(fluxrat	=	9.0878562927246)	Flux ratio threshold (in percent)
(npasses	=	5)	Number of detection passes
(window	=	5)	Size of detection window
(interac	=	yes)	Examine parameters interactively?
(train	=	yes)	Use training objects?
(objects	=	)	Cursor list of training objects
(savefil	=	)	File to save train objects
answer	=	yes	Review parameters for a particular image?
(mode	=	ql)	

You can edit the parameters for cosmicrays by scrolling down to the relevant parameter with the arrow keys, typing in that text field, and hitting Enter. Later, we will find that some parameters contain lists of sub parameters, but for this task in particular, there is only one level to the list.

Below, we highlight the most important parameters and explain how to determine their values. For our purposes, you can ignore the rest of the parameters, but you should check that they are at the defaults as listed on the previous page in non-bold text.

input	= *imagename*.fit		# Enter the image to be cosmic ray subtracted
output	= *new imagename*.fit		# Name of output image, maybe BO725c.fit
(thresho	=	100.)	# Or set to $\approx 5 \times \sqrt{MEAN}$, whichever is greater
(fluxrat	=	7)	# This is an initial flux ratio (in %), it can be changed
			# interactively in the next step. (usually 5-7% is standard)
(interac	=	yes)	# Make sure interactive is set to yes, to do it manually
(train	=	yes)	# Make sure to use training objects

When you have edited and checked the important parameters, type ":g" to run the task

Now a cursor will pop up on your image in the ximtool. First you identify the stars to give the program an idea of what not to detect as cosmic rays. To do this, you magnify a star with the middle mouse button and hit 's' on the center of the star. Do this with at most 10 stars and then move onto the cosmic rays. Here you also magnify the image and then hit 'c' on the cosmic rays, also doing no more than 10. When finished, hit 'q' and a Tek window will pop up with a plot of flux ratio vs. flux that looks like the plot on the left below:

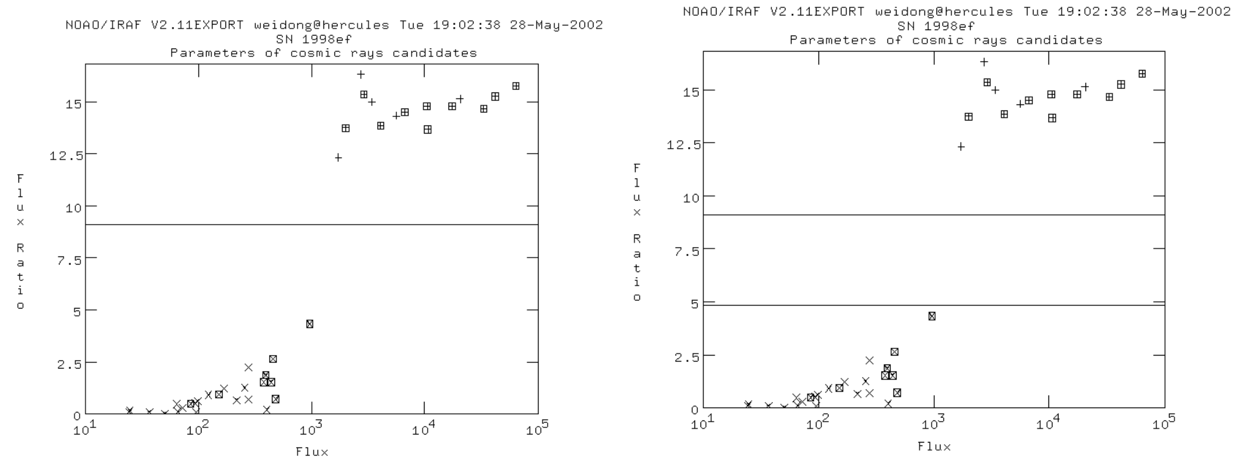

Fig. 11.— Plot of flux ratio vs. flux, for some objects in the image. Those detected as cosmic rays are marked with an "×". Those detected as stars are marked with a "+". Stars and cosmic rays that you identified by hand also have a box around the "+" or the "×". In the second plot, we have adjusted the flux ratio to a lower value by hitting 't' in the Tek window. Thus the second lower horizontal line.

Now you will be asked on the xterm command line if you wish to review the parameters. If you enter "yes" you can adjust the flux ratio cutoff interactively by putting your cursor on the horizontal line that denotes the cutoff between stars and cosmic rays, then hitting 't' in the Tek window at the spot where you want a new horizontal line to appear and denote the new flux ratio.

When you are done, hit 'q' in the TeK window, and the cosmic ray subtraction has been performed.

To understand how this worked, we have to discuss what the flux ratio actually is. The Flux Ratio itself is just a quantity for a given pixel that compares the number of counts in that pixel to some average of the number of counts in the nearby pixels. It takes the ratio of the flux in the nearby pixels to the flux of the pixel in question. Cosmic rays can yield pixels with a high number of counts, but just having a high number of counts in a pixel is not a sufficient condition for calling that pixel a cosmic ray since that pixel could also be on a star and still have a high number of counts. Having a large number of counts is thus only a necessary, but not a sufficient condition, for that pixel to be a cosmic ray. So in order to distinguish between stars and cosmic rays, we take advantage of the fact that cosmic rays are often isolated to just single pixels or a few pixels, and use the flux ratio as a measure of how the counts in a pixel compare to its neighbors. For a cosmic ray, the flux ratio is likely to be low, since the surrounding pixels will generally have a lower flux than the cosmic ray pixel, whereas for a pixel in a star, the flux ratio is likely to be higher, (closer to 1), since light from the star is spread out over and large number of pixels, causing adjacent pixels to have similar numbers of counts.

If, for example, you made the flux ratio too high, it would regard essentially everything in the image as a cosmic ray, and delete it. To avoid this, usually, you never exceed a flux ratio of 10% and most flux ratios be below 5% are sufficient to detect most cosmic rays. And like dark current subtraction, this process need not be perfect. We are simply cleaning the images to improve the accuracy of photometry, but this accuracy can never be perfect. And in general, cosmic rays only become a problem that might skew some of the data points when they land too close to the pixels with the supernova or standard star. This happens, but is fairly rare, so most of the time, for photometry purposes, cosmic ray subtraction ends up being a precaution.

Once we have done this manually for all the images with high MEAN, we can do it automatically for the rest of them. First create a list of the images to be done automatically if you haven't already done so (Blist_auto). Now edit the parameters for cosmicrays and make the following changes that differ from doing cosmic ray subtraction manually.

cl> epar cosmicray # edit the important parameters (epar) of the IRAF task cosmicrays,

input	=	@Blist_auto	# Enter "@" and the list of images to be done automatically
output	=		# Enter nothing, unless you create a list of output images. If not, it
			# will overwrite the images with ones that have been cosmic ray subtracted
(thresho	=	100.)	# Same threshold for all images
(fluxrat	=	7)	# This is the default flux ratio from the last image you did manually.
			# in general, a flux ratio of 5-7 percent is reasonable
(interac	=	no)	# Make sure interactive is set to no
(train	=	no)	# Do not use training objects
			# Or else you'll just be doing it manually.

Again, for our purposes, you can ignore the rest of the parameters. When you have edited and checked all the important parameters, again, to run the task, type ":g" to run the task. It should run basically instantaneously. And unless you specified a list of output images that you created, it will overwrite the images with ones that have been cosmic ray subtracted automatially. But generally, this is safe. If you're worried you can create an output list. Repeat this process for every band. In the end, create a list of all the cosmic ray subtracted images and display them to see if you have gotten most of the cosmic rays

Now if everything worked, you're done. But first, inspect the images to see if any stil require some manual cosmic ray subtraction. For individual image comparison, you can display the original image in frame 1 of the ximtool and the comsic ray subtracted image in frame 2. To blink between frames to clearly see the differences between them, you can hit "Ctrl F" in the ximtool window. You should be able to tell how well the cosmic ray subtraction has worked and whether it is worth it to redo any of the images manually by adjusting the flux ratio and/or the threshold of detection.

In the program, we set the threshold of pixel counts below which you should disregard pixels as potential cosmic rays, and run the task. The task cosmicray then measures the flux ratio for all the pixels, and then detects what it thinks are a bunch of cosmic ray candidates and displays them schematically in the Tek window, as you have seen. Now this process may fail if the cosmic ray is lower energy and thus registers fewer counts than the threshold (or has a lower flux ratio than what we choose), compared to its neighbors, or if the cosmic ray spreads itself out over enough pixels to appear starlike.

For large cosmic rays that are sufficiently starlike (or close enough to the supernova or a standard star) that are missed by the IRAF cosmicrays program entirely, you can detect them by eye and remove them manually with the IRAF command imedit. Make sure the image is open in an ximtool and in IRAF, type:

cl> imedit *imagename* imagename # edit the pixels in the image manually

Now use the middle mouse button to magnify the cosmic ray, and to delete individual pixels (or, rather, match their values to nearby background pixels).[27] You can do this either by hitting 'c' above and below a cosmic ray pixel (or above and below a vertical column of cosmic ray pixels), or hitting 'l' to the left and to the right of a cosmic ray pixel (or a horizontal row of cosmic ray pixels). Be careful if the cosmic ray is too close to the supernova or the standard star. If this is the case, do not delete the cosmic ray, just make a note of it and, at the worst, case that image would have to be thrown out, but chances are it will still be OK to use for photometry, and you'd actually have cause a larger photometric error by deleting the cosmic ray, and probably also removing some of the supernova or standard star light. In any case, now you have a new image that has been cosmic ray subtracted. Move the old image and rename the new one as you see fit.

[27] imedit only works on images with a a given pixel type called "real", so if imedit fails, you may have to change the pixel type with the "chpixtype *imagename*" command in IRAF. It will ask you some questions and you want to change the pixel type to "real". Afterwards, try imedit on the new image and see if it works.

6. Photometry: Reducing the Data

Now that the images have been prepared and cleaned, we can get onto the actual data reduction, and do some photometry. This is the major section of this primer. Unfortunately, it is a rather detailed process, so it will take quite a while to cover. Basically, there are 3 major steps: Galaxy Subtraction, Aperture Photometry, and PSF photometry. You only need to perform galaxy subtraction on images where the supernova occurs close to the nucleus of its host galaxy, otherwise you can move on directly to aperture photometry. You need to use PSF photometry mainly on supernovae which have a crowded field of stars in the image or occur close to the nucleus. You do aperture photometry on all images, because it is both a standard photometric technique and a precursor to PSF photometry, which you may or may not do. In this section, we will explain what each of these tasks are and how to perform them in detail using IRAF software packages. Once you finish galaxy subtraction, aperture, and psf photometry, you have all the instrumental magnitude data and all you will need to do is display it in meaningful form and transform to standard magnitudes, which will be covered in the next major section.

6.1. Galaxy Subtraction*

Essentially all supernovae will occur in elliptical or spiral galaxies, which can be composed of as many as $10^{11} - 10^{13}$ individual stars.[28] A good rule of thumb is that, on average, for each galaxy, there is a supernova of some type every 100 years or so. The particularly amazing thing about supernovae is that is that, despite being overwhelmingly outnumbered by the other stars, individual supernovae can sometimes outshine their entire host galaxy. Now this is very impressive, but unfortunately, the light from the galaxy itself is still pretty bright even when compared to the supernovae light, and thus can interfere with our efforts to do photometry. In essence, we want to count the light from the supernova and the supernova alone, so if light from other stars in the galaxy lands on the same pixels as the supernova (which it inevitably will), we have to remove it to extract a genuine star signal. This is not so different from removing dark current and cosmic rays, since for our purposes, photons from other stars in the host galaxy are simple not what we are interested in here. In the end, the process by which we remove the galaxy light from our images, leaving only the light from the supernova and standard stars, is called galaxy subtraction.

When we actually do the subtraction, we want to take an image of the galaxy without the supernova in it, and subtract it from one of our images that has the supernova in it.[29]. In theory, this will leave only the light from the supernova, which is what we are interested in. What this means is that you have to have a high quality template image of the host galaxy where the supernova

[28]Individual stars or binaries that escape from galaxies and end up in intergalactic space could conceivably undergo supernova, but the chances of our finding one not in a galaxy is so small as to be basically negligible.

[29]What we actually do, as to not interfere with the standard stars outside the pixels of the host galaxy is to subtract the galaxy light from a 80 x 80 pixel or so search box surrounding the supernova, and then paste the subtracted piece (just the supernova) back onto our original image at the same coordinates. But more about that later.

is not in the image. There are two possible ways to get this image. If we just so happen to have taken good quality template images of the host galaxy before the supernova explosion, then the process is straightforward. Unfortunately, in most cases, we do not have such a template, since we can't devote our time to taking calibrations images of every galaxy that KAIT observes, because most of them never yield supernova and we would just be wasting our time. This means that for most supernovae, we must wait 6 months to a year until the supernova fades away before we can take a calibration image of the host galaxy on a very clear night. Since the supernova is no longer visible in this image, we can use it as the galaxy template to subtract from our supernova images.

Getting good galaxy templates actually represents a serious bottleneck to our data reduction efforts, since we hope to generate the light curves for a large number of Type Ia supernovae (over 30) who we have followed extensively with KAIT, but whose data are simply waiting idly by while we try to get good calibration images of the host galaxy. We have to wait a long time both for the supernova to fade and for a sufficiently clear night, which means we have to wait a very long time to get a calibration image. Without this template image, for supernovae that require galaxy subtraction, we can't even begin to do any of the photometry. The long timescale to complete a light curve should be evident as you find yourself working with data for supernovae that were discovered as early as 1998. We have an embarrassment of riches, but with your help as part of a team of students available to help reduce the plethora of data we have, we finally have the motivation to get the necessary templates, and produce some long awaited light curves.

6.1.1. When Do You Actually Need Galaxy Subtraction?

So when do you actually need to do galaxy subtraction? As it happens, you only need to perform galaxy subtraction if the supernova occurs close to the nucleus of its host galaxy. (hence the asterisk * at the beginning of this section.) If it occurs far away from the nucleus toward the edge of the galaxy (i.e sn2000cx, sn2002ap), you can proceed directly with aperture and PSF photometry as described in the next section . The reason for this is that the number of stars in the galaxy, and hence the amount of galaxy light, falls off exponentially as you move from the center toward the edge of the galaxy. Most stars lie near the central nucleus of the galaxy and fewer reside toward the edges, so if your supernova occurs towards the outskirts, we can approximate that the light from the galaxy at that place in space is negligible compared to the light from our supernova, and not worry about galaxy subtraction.

Unfortunately this rarely happens, since more stars exist closer to the nucleus, making the probability of discovering a supernova near the nucleus much greater than the probability of finding one at the edge of the galaxy, based simply on the number of stars in these regions that could potentially give rise to a supernova. As a result, most supernovae will require galaxy subtraction in order to get adequate photometry. But there are a few ones we find (such as sn2000cx) that are far enough away from the nucleus to not require galaxy subtraction, and those are usually the ones you would practice on when learning aperture photometry. Some examples of galaxies with supernovae which may or may not need galaxy subtraction are shown on the following page.

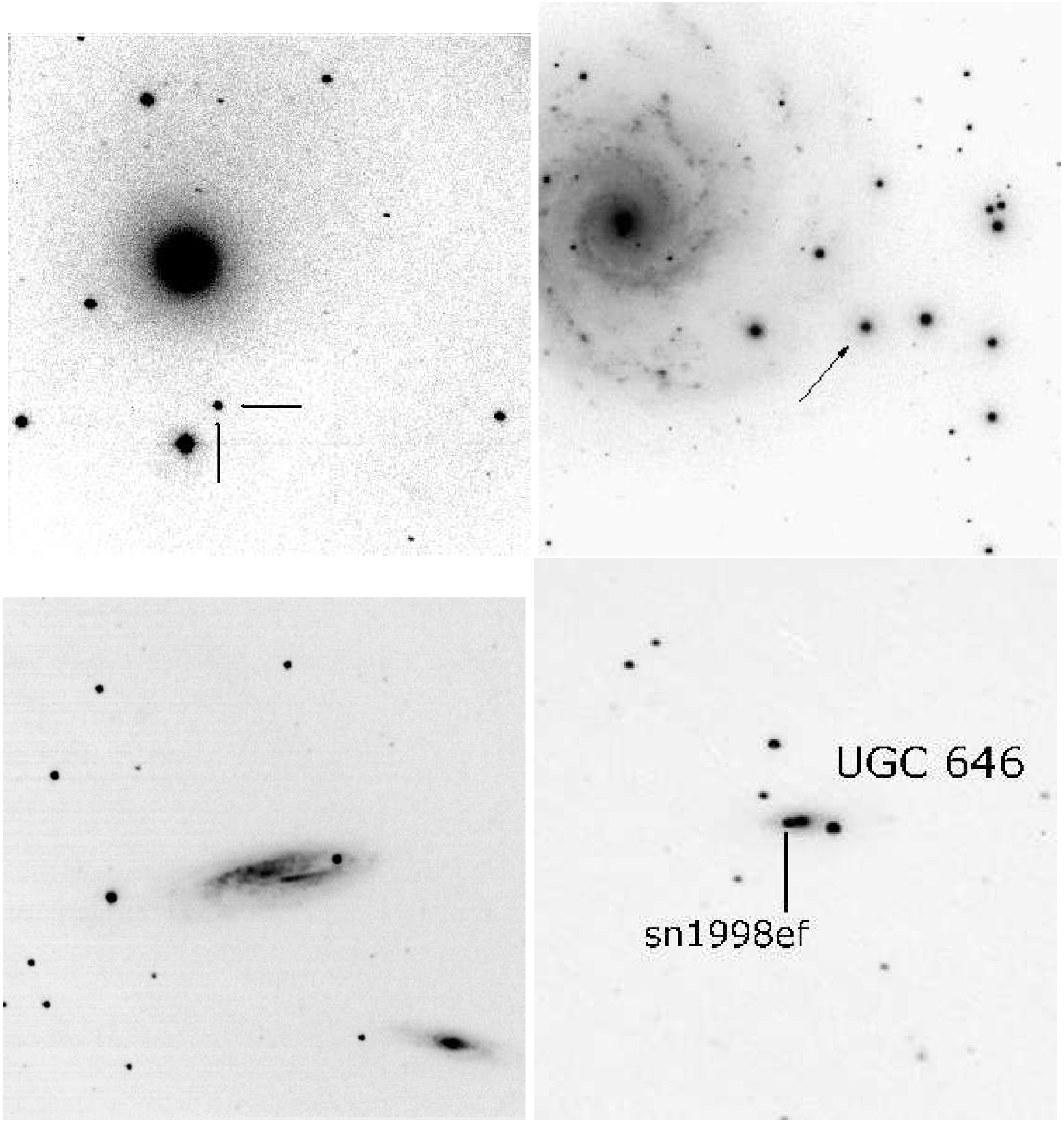

Fig. 12.— Some examples of supernovae that do or do not require galaxy subtraction. **Row 1**: (left to right) Supernovae 2000cx and 2002ap: They don't need galaxy subtraction because they occur far enough away from the nucleus of their host galaxy. Supernova 2000cx only required aperture photometry, while 2002ap also needed PSF photometry. **Row 2**: (left to right) Supernovae 1998dh (in upper right corner of host galaxy) and 1998ef: They do need galaxy subtraction because they occur close to the nucleus of their host galaxy. sn1998ef in particular is practically buried in the nucleus, ensuring that the galaxy light will be comparable to the supernova light in that region on the CCD image. Both these supernova required both aperture and PSF photometry.

6.1.2. Performing Galaxy Subtraction Manually

If you decide that the supernova is close enough to the nucleus of its host galaxy and that you need to perform galaxy subtraction, the first thing you must do is to find the galaxy template image (or images) in your directory or get them from Weidong. In principle, all you need is one good galaxy template, but occasionally, we are able to take two calibration templates, so that you can do photometry on both and compare to see which template yields better results. If you get an initial directory of data where the dates of observation in the filenames all come from, say, 1998, and then you find an isolated image from 2000, and maybe another from 2001, you can be pretty sure that the latter two are galaxy template images. To be sure, just display them and confirm that the supernova can not be seen in the image.

Once you've found the relevant template or templates, you must use an IRAF task called **automktem_t**, in order to create a standard template that can be used for the relevant galaxy subtraction software. Just as with cosmic ray removal, the IRAF task has a list of parameters to be edited before you can run the task and create the template. The list of parameters and the most important ones to be edited for automktem_t can be found two pages from now.[30]

Once you've created your galaxy template or templates, you can begin to use the relevant galaxy subtraction software. There are two sets of software, or "pipelines" that we use for galaxy subtraction. One is Weidong's pipeline, and the other is the High-Z pipeline. Both are very good at galaxy subtraction, and generally yield very similar results, however there is some indication that Weidong's pipeline is superior for poorer quality templates (worse seeing during the observation) and the High-Z pipeline works fine for good quality templates. In principle, one could use both pipelines for each band and generate 8 light curves (16 if you did it for 2 galaxy templates), but this seems like a bit of overkill. In addition to being overkill, this would introduce a bias because you would likely choose whichever light curves that looked the best and had the least amount of scatter in the data points, to see which yields better results.[31] Nevertheless, it is useful to try doing photometry with both pipelines in at least one band, to help choose which pipeline to use for the rest if the bands.

So, in general, what might be useful is to try doing photometry with the output images from both galaxy subtraction pipelines, in say, the B band only. Then generate an instrumental light curve and look at the results to see which pipeline looks better to use for the rest of the bands. Sometimes galaxy subtraction will fail, so in general, it seems reasonable to choose the pipeline

[30] When you eventually run the galaxy subtraction software automatically with one of Weidong's IRAF scripts called tempsub.cl, it automatically aligns, or registers the images for you so that all the stars have the same x-y coordinates in all the images. If you do not need galaxy subtraction, you will need to register the images later when you do aperture and PSF photometry automatically. In that case, you would have to run automktem_t on a normal image with the supernova and use that as a template to register all the other images around using one of Weidong's programs called register.cl But in either case, you will only need to run automktem_t once, either to create the template for galaxy subtraction, or to create a template to register the images before doing automatic aperture and PSF photometry.

[31] One possible way around this would be to average the data from the 2 pipelines.

that yields the most good subtractions. If the better pipeline happens to fail for a few days, while the worse one happens to succeed for those days, it is reasonable to borrow the good subtractions and use them to help generate your final light curve. But is is not a good idea to do this for more than 2 or 3 images since you could introduce biases into your results by not having a uniform data analysis technique on all the images. But for the most part, the two pipelines seem to give fairly consistent results to high accuracy, as demonstrated by many experiments Weidong has performed on the data.[32]

In a situation where two people are doing photometry on the same supernova and there happen to be two galaxy templates, rather than duplicate work, it would be useful to have one person use a unique template and pipeline set. Weidong and I did this for supernova 1998dh and, even with different galaxy templates and galaxy subtraction pipelines, our results were quite consistent, providing an excellent cross-check of the statistical robustness of the data analysis. But in general, you should figure out which pipeline to use in advance depending on the details of your particular data set, and by trying two pipelines at least for a single band, to see which pipeline yields more good subtractions. Some examples of galaxy templates are shown below on the left and middle.

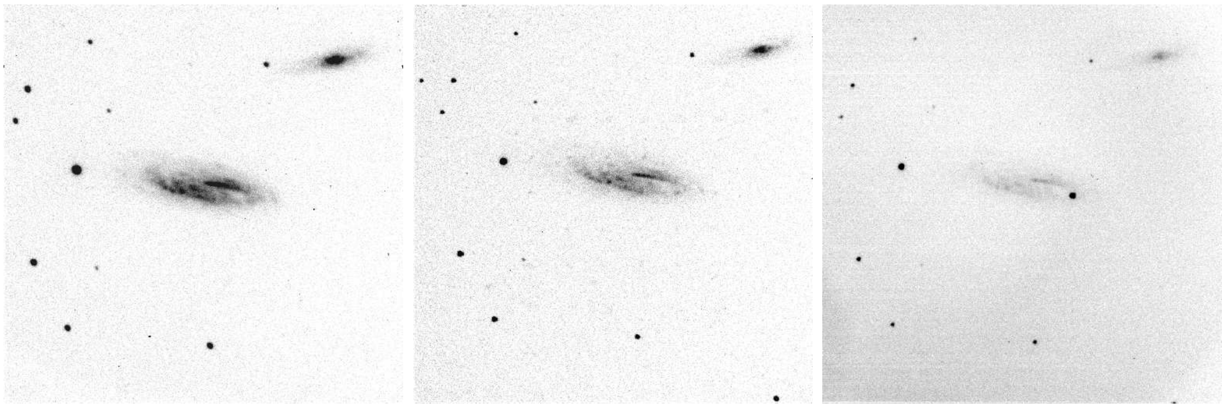

Fig. 13.— (Left and Middle): Two example galaxy template calibration images for supernova 1998dh, where the supernova is not apparent. Both images were taken well after the supernova has become too faint to see (left: 9/25/2000 and middle: 7/19/2001), and can thus be used for galaxy subtraction. The final image on the right, taken on 8/7/1998, shows the supernova, (in the lower right corner of the host galaxy) for comparison.

On the next page, we discuss how to take these template images and use them to create standard templates which can be used for both galaxy subtraction pipelines.

[32]If you're just learning, you can conceivably go through all photometry steps for the B band first, then do the other bands. Or you can do them all at the same time. Either way, its up to you.

da> epar automktem_t # edit the parameters for the IRAF task **automktem_t**

I R A F
Image Reduction and Analysis Facility

PACKAGE	=	user	
TASK	=	automktem_t	
inputA	=	B98dh.imh	image name
fieldnam	=	B98dh	name of field
delnot	=	no	Delete fit after mktemp?
(three	=	no)	Combine 3 files
(tmin	=	10.)	min thresh for dophot
(tmax	=	10000.)	max thresh for dophot
(FWHM	=	2.83)	Approximate FWHM of images
(saturat	=	50000.)	level for saturation
(oblit	=	4.)	obliteration size
(smooth	=	no)	Do you want to smooth image
(smoothv	=	1.5)	amount to smooth (FWHM)
(dof	=	yes)	Is this for findsn?
(fwfile	=	/cetus/weidong/temp/ssigma.dat)	
(shfile	=	sexshift.dat)	
(galfile	=	/whale/weidong/template/F20918.gal)	
(mode	=	ql)	

inputA	=	B98dh.imh	# input image name used to create template
fieldnam	=	B98dh	# name of field, usually filter and supernova is a good idea

Table 7: Again, here are the most important parameters and how to edit them. When you're done editing the parameters, hit ":g" to run automktem_t

After it runs, automktem_t will spit out these files.

B98dh.fit	# The actual template image
B98df.fwhm	# Contains same information as from hitting 'a' on stars in imexam mode (i.e GAUSSIAN)
B98dh.info	# contains important information that you may need to modify later
B98df.stars	# I don't really know what this is...ask Weidong

When you've finished creating the galaxy template or templates using automktem_t, you now need to run the galaxy subtraction software manually on at least one image. What you are essentially doing here is using a program of Weidong's called **fsn2** which first registers the image you are working on to the same coordinates as the template, and then selects a square n x n pixel search box (usually n = 80) centered around the supernova, and subtracts the template image from the supernova image. The result should then be an image with a box that essentially contains light from the supernova alone. But the standard stars around the box may have been screwed up by the galaxy subtraction step, so what you now want to do is paste just the search box with the supernova back onto your original image, which just overwrites all the pixels at the search box coordinates in your original image. You now have an image with a box containing just the supernova (and no galaxy light), and the standard stars in the field unchanged by the galaxy subtraction step. This image is then ready for aperture and PSF photometry, as described in the next sections.

Assume, for example, that the galaxy template is named B98dh.fit, and that the image you are working on is named B0729.fit. Assume, also, for example, that the coordinates of the 80 x 80 pixel search box in the supernova image and the galaxy template (after registering) are given by [$xleft$:$xright$, $ybottom$:$ytop$] = [287:367, 228:308].[33] This coordinate list is called the template "section". You can find the template section rather easily by finding the x-y coordinates of the supernova as it would have appeared in the galaxy template. Just display the galaxy template in the ximtool and move the cursor to where the SN would have been, and notice the x-y coordinates displayed in the lower left corner of the ximtool. Now offset these coordinates by ± 40 in all 4 directions to generate the list [$xleft$:$xright$, $ybottom$:$ytop$]. As long as the supernova is near the center, you're fine, it only needs to be approximate. You can also change the size of the search box to, say, 60 x 60 for galaxies that subtend fewer pixels in your image. Anyway, here's how to do galaxy subtraction manually on an image in IRAF. Make sure an ximtool is open first.

cl> cp B0729.fit B98df_001.fit	# copy your image to a new filename of the form *template name*_001.fit
	# fsn2 always takes as input images with that same filename format
cl> fsn2 B98df_001.fit B98df	# Run fsn2 on image B98df_001.fit, using galaxy template name B98df

fsn2 executes the major galaxy subtraction step, it registers the supernova image to the same coordinates as the galaxy template image, then outputs the following images from each pipeline:

observation.imh	# Your supernova image, converted to a .imh file (both pipelines)
him_shift.sub.fits	# subtracted image in .fits format (High-Z pipeline)
subtracted.imh	# subtracted image as a .imh file (Weidong's pipeline)

Now let's deal with Weidong's pipeline first, starting on the next page.

[33]Notice that $xright - xleft = ytop - ybottom = 80$.

```
cl> disp subtracted.imh 1     # Display subtracted.imh in frame 1
cl> disp observation.imh 2    # Display observation.imh in frame 2 (can blink frames with Ctrl-F)
cl> imexam                    # Hit 'm' to find sky background (MEAN=x) near the
                              # supernova in observation.imh
```

You find the mean sky background (MEAN=x) near the supernova in the observation because you want to make sure that you don't get any pixels with negative values for counts in the subtracted image. This could easily happen if the sky background (which fluctuates even on very short timescales), happened to be brighter in the galaxy template than in the observation. To correct for this, we use the imarith command, just like for dark current subtraction, but this time we're just adding a constant offset to every pixel in the subtracted image.

```
cl> imarith subtracted.imh + x subtracted.imh   # Offset subtracted image by x and
                                                # overwrite old subtracted.imh
```

This resets the O scale to avoid pixels with negative counts counts in subtracted.imh. Now we want to copy the search box in the subtracted image with the same coordinates as the template section [287:367, 228:308], and then paste it back onto the observation at those same coordinates. This can be done, pixel by pixel, with the imcopy command.

```
cl> imcopy subtracted.imh[287:367, 228:308] observation.imh[287:367, 228:308]
```

Now display observation.imh again, and the search box with the supernova should stand out.

```
cl> disp observation.imh 3                  # Display observation.imh in frame 3. You can compare
                                            # it to original observation.imh, still in frame 2
cl> imrename observation.imh s2_B0729.imh   # Rename observation.imh to original filename of supernova
                                            # You use the imrename command instead of mv
                                            # for .imh files for same reason you don't use rm command
                                            # The image header and .pix file are stored separately
```

You want to rename observation.imh to a new filename since the next time you run galaxy subtraction it will get overwritten. You want to use the s2 prefix because in our file naming system, Weidong's pipeline goes by s2 and the High-Z pipeline goes by s1.

s1	High-Z pipeline
s2	Weidong's pipeline

The naming scheme has to do with the way the output files are named when you eventually run the IRAF script which will do galaxy subtraction automatically. But now that you have the file s2_B0729.imh, you've finished galaxy subtraction on that image for Weidong's pipeline and the image is now ready for aperture and PSF photometry.

Now we need to show how to do galaxy subtraction manually for the High-Z pipeline. It's a bit more complicated because it puts the subtracted image into a different subdirectory called sexdir[34], but otherwise, the steps are the same as for Weidong's pipeline, just substituting "subtracted.imh" with "him_shift.sub.fits". Make sure you're in IRAF, and that you've got an ximtool open.

```
cl> cp B0729.fit B98df_001.fit     # copy your image to a new filename of the form template name_001.fit
                                   # fsn2 always takes as input images with that same filename format
cl> fsn2 B98df_001.fit B98df       # Run fsn2 on image B98df_001.fit, using galaxy template name B98df
                                   # Now here's the difference...
cl> cd sexdir                      # Change to the Star Extraction directory created by the High-Z pipeline
cl> cp him_shift.sub.fits ..       # Copy the High-Z subtracted image to the directory one level up
                                   # I've tried copying observation.imh to the sexdir directory, but I've
                                   # found that imcopy doesn't work in that directory for some reason
cl> cd ..                          # Go back up one directory, which you just copied the file to
cl> disp him_shift.sub.fits 1      # Display the High-Z subtracted image in frame 1
cl> disp observation.imh 2         # Display observation.imh in frame 2 of the ximtool
cl> imexam                         # Hit 'm' near SN in observation.imh to find MEAN sky background = x
```

Now adding a constant offset x to every pixel in the subtracted image.

```
cl> imarith him_shift.sub.fits + x him_shift.sub.fits    # Offset subtracted image by x
```

Use the same coordinates for the template section [287:367, 228:308], and then paste the supernova search box from the High-Z subtraction it back onto the observation at those same coordinates.

```
cl> imcopy him_shift.sub.fits[287:367, 228:308] observation.imh[287:367, 228:308]
```

Now display observation.imh, and again, the search box with the supernova should stand out.

```
cl> disp observation.imh 3                     # Display observation.imh in frame 3. You can compare
                                               # it to original observation.imh, still in frame 2
cl> imrename observation.imh s1_B0729.imh      # Rename observation.imh to original filename of supernova
```

Always rename observation.imh to a new filename since the next time you run galaxy subtraction it will get overwritten. This time, you want to use the s1 prefix because the High-Z pipeline goes by s1 in our file naming system. If you want to repeat the process for a different galaxy template you've created, you'd need to use a different template name (i.e. BB98df.fit), and a different 80 x 80 pixel template section (i.e. [266:346, 216:296]), because fsn2 registers the images to the coordinates of the template, and since separate templates are unlikely to have the same x-y coordinates.[35]

[34] which stands for **S**tar **Ex**traction **Dir**ectory, not what you might think at first.

[35] There is no guarantee that both templates will cover exactly the same field of view due to the inevitable slight differences in telescope pointing during the two template observations, which may be widely spaced in time.

That's it. You've now done galaxy subtraction manually with both pipelines and have created 2 images that are ready for aperture and PSF photometry. Now for clarity, here's an abridged version of manual galaxy subtraction for both pipelines. It excludes comments as to what each step means because I could not find a practical way to fit them all on the same table. If you forget what a step means, please refer to the previous pages.

cl> cp B0729.fit B98df_001.fit
cl> fsn2 B98df_001.fit B98df
cl> disp subtracted.imh 1
cl> disp observation.imh 2
cl> imexam
Hit 'm' to find MEAN = x
cl> imarith subtracted.imh + x subtracted.imh
cl> imcopy subtracted.imh[287:367, 228:308] observation.imh[287:367, 228:308]
cl> disp observation.imh 3
cl> imrename observation.imh s2_B0729.imh

Table 8: How to do galaxy subtraction manually on an image with Weidong's pipeline (s2)

cl> cp B0729.fit B98df_001.fit
cl> fsn2 B98df_001.fit B98df
cl> cd sexdir
cl> cp him_shift.sub.fits ..
cl> cd ..
cl> disp him_shift.sub.fits 1
cl> disp observation.imh 2
cl> imexam
Hit 'm' to find MEAN = x
cl> imarith him_shift.sub.fits + x him_shift.sub.fits
cl> imcopy him_shift.sub.fits[287:367, 228:308] observation.imh[287:367, 228:308]
cl> disp observation.imh 3
cl> imrename observation.imh s1_B0729.imh

Table 9: How to do galaxy subtraction manually on an image with the High-Z pipeline (s1)

That's it for manual galaxy subtraction. An example of some images before and after galaxy subtraction are shown on the following page. After that, we can move on towards doing galaxy subtraction automatically using an IRAF script written by Weidong called tempsub.cl

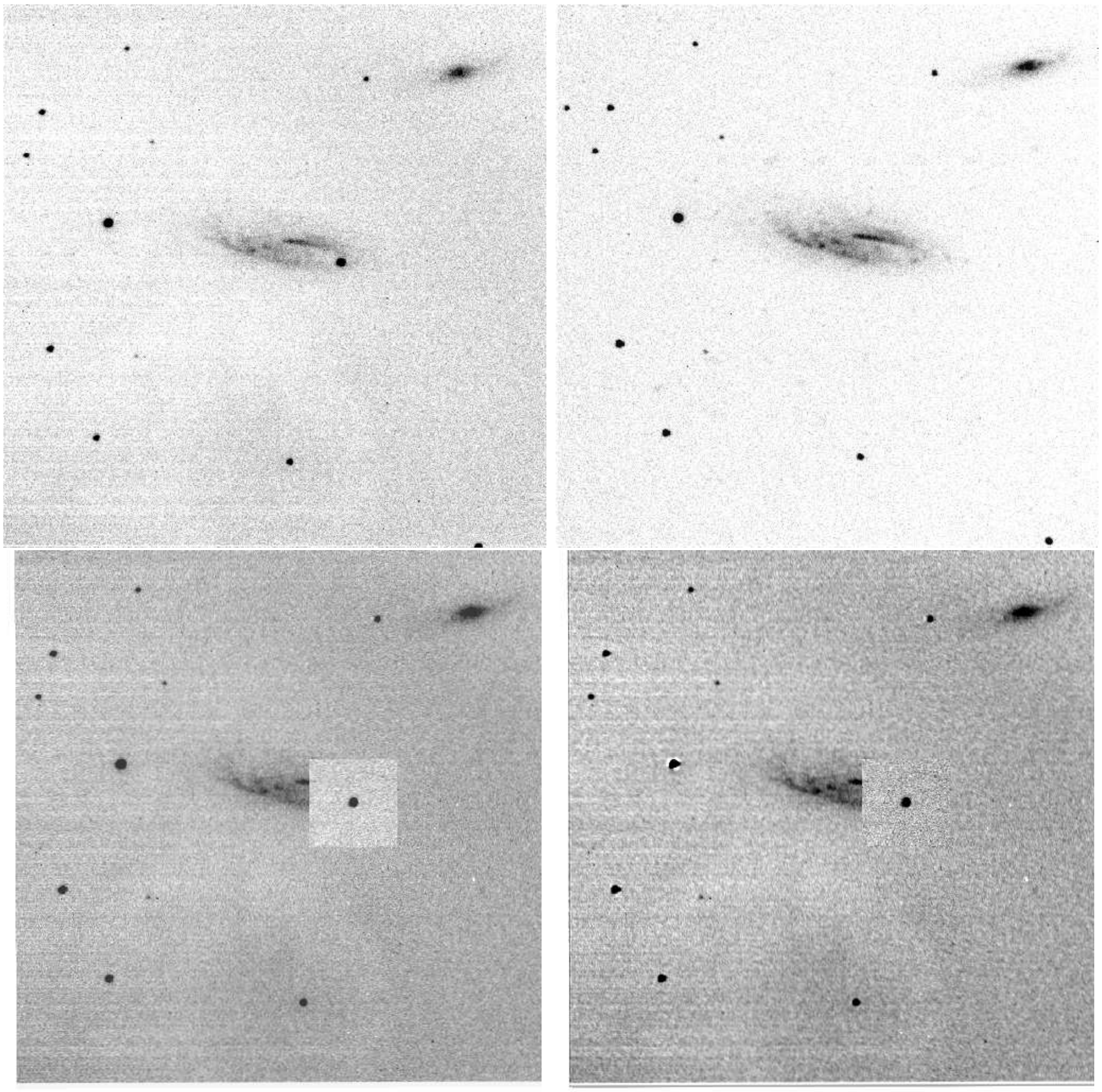

Fig. 14.— An example image from supernova 1998dh taken on 8/12/98 is shown going through the galaxy subtraction process with both pipelines, each using a galaxy template image taken on 7/19/2001. **Upper left**: The original supernova image. **Upper right**: The galaxy template where the supernova is not in the image. **Lower left**: The galaxy subtracted image of the supernova obtained using the High-Z pipeline. **Lower right**: The galaxy subtracted image of the supernova obtained using Weidong's pipeline. Notice the easily apparent search boxes surrounding the supernova with galaxy light removed. Notice also that all images are 500 x 500 pixel squares of the same size and that both the galaxy subtracted images appear to have an L-shaped section missing on their lower left (which just appears white). This occurs because the galaxy subtraction program fsn2 registers the images to the same coordinates as the galaxy template, which had a slightly shifted field of view. Note that if a different template was used, the shift to register the images might be different and the search boxes might appear at different [*xleft*:*xright*, *ybottom*:*ytop*] coordinates, for example [266:346, 216:296], rather than [287:367, 228:308] as in the above images.

6.1.3. Performing Galaxy Subtraction Automatically

Now fortunately, you won't have to do galaxy subtraction manually on all the images since that would take forever. In this spirit, Weidong has written an IRAF script which essentially does all the steps that you just did automatically for both pipelines. First, you need to get a copy of tempsub.cl and put it in the directory with the image you want to run galaxy subtraction on. Then you need to create a list of files to do galaxy subtraction on. In the B directory, for example, use the "ls B*.fit > blist" command. Now you'll need to edit blist using vi or emacs and make sure to remove the image you already did galaxy subtraction for manually, and remove any .fit files from the list, such as the galaxy templates themselves (i.e. B98df.fit, BB98df.fit), and the old input to fsn2 (B98f_001.fit). When the list includes only the files you wish to perform galaxy subtraction on, now you'll need to actually modify a few things in the IRAF script, an example of which is shown on the next page. When viewing the actual code of tempsub.cl on the next page, you can see a direct correspondence between everything you typed on the command line for manual galaxy subtraction. This is essentially all that a script is, a time saver for repetitive tasks. You'll need to edit the script using vi or emacs, and change a few relevant parameters:

list = "blist"	# or whatever you named the list of images to register
v99el	# anywhere it says v99el, change it to our galaxy template name (i.e B98dh)
[165:255, 273:333]	# Change the template section wherever it appears to the coordinates
	# for your galaxy template, i.e. [287:367, 228:308] in our earlier example

In principle, this program could be written in such a way as to allow you to change these parameters by being prompted to type things in at the command line, but for now, you can just keep a separate copy of tempsub.cl in each of your directories for all the bands, and make minor modifications to the script as needed. Once you're in the proper directory and ready to go, in IRAF, with an ximtool open, define tempsub.cl as an IRAF task by typing:

cl> task $tempsub=tempsub.cl	# defines tempsub.cl as an IRAF task to run by typing "tempsub"
cl> tempsub	# This will run galaxy subtraction on all the images in blist

The first image named in list should now pop up in the ximtool. The only thing you'll need to do as the program is running is enter the mean of the sky background of the observation near the supernova. You can just move the cursor there and look at the counts displayed in the lower corner of the ximtool to get a rough idea. Enter that number x, and hit a bunch of carriage returns (16, I believe), and a subtracted image with search box will show up in the ximtool. Hit 'q' in the ximtool to move on to the next image. If the program ever outputs an error message, like "warning, Gaussian fit did not converge' and the program halts, just go back and modify blist to remove the names of images you've already done, make sure the current image is copied to the filename *template name*_001.fit, and then run tempsub again. That will normally fix 90% of the galaxy subtraction errors. For other errors, keep a list and ask Weidong for help.

```
# Rename list to be the new filename
  list="blist"
  while(fscan(list, s1)!=EOF)
  {
   display(s1, 1)
# Change V99el to the template name
   if(access("V99el_001.fit")) delete("V99el_001.fit")
# Change V99el to the template name
   print("cp "//s1//" V99el_001.fit") — cl
# This is the real galaxy subtraction step. If an info.old file exists, use it.
   fsn2("V99el_001.fit", "V99el", autorep-)
# This step display the observation and subtraction, and choose
# an offset (the background of observation) between them
   display("subtracted", 2)
   display("observation", 1)
# Blink them with Ctrl + F
   printf("Input the background of the observation:")
   i=scan(x)
   if(access("s1.imh")) imdel("s1")
   if(access("s2.imh")) imdel("s2")
   if(access("obs1.imh")) imdel("obs1")
   if(access("obs2.imh")) imdel("obs2")
# The High-Z subtraction is in s1
# The Weidong subtraction is in s2
   imarith("sexdir/him_shift.sub.fits", "+", x, "s1")
   imarith("subtracted.imh", "+", x, "s2")
# Paste section of the SN back to the observation
# Change the numbers for the section!!!!!!
   imcopy("sexdir/him_shift.conv.fits", "obs1")
   imcopy("observation", "obs2")
   imcopy("s1[165:255, 273:333]", "obs1[165:255, 273:333]")
   imcopy("s2[165:255, 273:333]", "obs2[165:255, 273:333]")
# Display and compare them
   display("obs1", 1)
   display("obs2", 2)
   imexam
# The final results are saved in s1_*.imh ==> high-Z subtractions
#                                  s2_*.imh ==> Weidong subtractions
   imrename("obs1", "s1_"//substr(s1, 2, stridx(".", s1)-1))
   imrename("obs2", "s2_"//substr(s1, 2, stridx(".", s1)-1))
}
list=""
```

Table 10: Example code for IRAF script **tempsub.cl**

6.2. Aperture Photometry

For a supernova that occurs far from the nucleus of its host galaxy in a relatively uncrowded starfield, the technique of aperture photometry will be sufficient to generate accurate light curves. In this section we will discuss what aperture photometry is and how to do it manually using software in IRAF. Eventually, we will learn how to do aperture photometry automatically with an IRAF script, but first we must learn what it is and how to set up the parameters in IRAF. But first we have to discuss the difference between differential photometry and absolute photometry.

6.2.1. Differential Photometry

Photometry is the precise measurement of the brightness of astronomical objects. Generally, we are talking about measuring the brightness of point sources such as stars. For practical purposes, I will use the word star to mean any astronomical point source, including a supernova. What we find is that it is very difficult to measure the brightness (in magnitudes) of an astronomical object directly on a magnitude scale which all observers can agree upon. But when this is done, we call it **absolute** photometry. Most of the time, however, we measure the brightness of an object indirectly by getting some instrumental measure of its magnitude which is not observer independent, but depends on our measuring apparatus. We then compare that instrumental magnitude to standard stars of known magnitude where absolute photometry has already been performed, and, ultimately, use these standard stars to calibrate our object's brightness and put it on an observer independent scale. This is the method of **differential** photometry. So the first thing to talk about it how we even get a measure of the instrumental magnitude, which will eventually involve a basic discussion of aperture photometry.[36]

To measure brightness quantitatively, we want to know how many photons from our star are incident on our detector, per unit time per unit area, and convert this number to astronomical magnitudes. However, as was implied earlier, this number is often very difficult to measure directly. What we can measure is the number of photons, N, that are actually detected by our CCD in a given exposure time, t, in a given frequency band determined by your filter. By measuring these numbers again for a star of known magnitude under identical observational conditions, we can then apply the technique of differential photometry to determine the brightness of our star.[37]

To measure apparent brightness, we want to know the flux density F_ν, i.e how much energy is hitting our detector per unit time, per unit area, in the relevant frequency interval, and convert this number to magnitudes. From the equation below, if we knew the distance to the star, and

[36] Here we are talking about measuring the star's apparent brightness as seen from earth, and not its intrinsic brightness or luminosity. However, if we know the distance to the star, we can infer the star's luminosity from its apparent brightness using the inverse square law of light.

[37] Astronomers have done absolute photometry on a number of standard stars and put them in a catalog that other astronomers can then use as a database for differential photometry. The most famous such catalog is the Landolt standard stars, named after the astronomer who did the pioneering work.

its luminosity L_ν (i.e power per unit frequency interval), we could measure knowing the star's flux density F_ν from the equation:

$$F_\nu = \frac{L_\nu}{4\pi d^2} \text{ watts m}^{-2}\text{ Hz}^{-1}. \tag{1}$$

However, if we do not know its distance or luminosity, which is often the case, we can also measure the flux density directly with the following method. For a detector of area A, frequency band $\Delta\nu$, and photon detection efficiency (or quantum efficiency) η_ν, that measures N photons in a time t, we can also express F_ν approximately as:

$$F_\nu \approx \frac{Nh\nu}{At\Delta\nu\eta_\nu}. \tag{2}$$

However, as noted earlier, this number is often very difficult to measure directly.

The difficulty arises because the quantities A (detector area), $\Delta\nu$ (frequency interval of your filter), and η_ν (quantum efficiency in that band) are rather difficult to measure in practice. Measuring all the variables is tantamount to performing absolute photometry. So instead of measuring, A, $\Delta\nu$, and η_ν, what we can measure directly quite easily are the remaining two quantities N and t, the number of photons N that are detected in a given exposure time t for a given filter. By measuring these numbers again for a star of known magnitude (such as a Landolt standard) under identical observational conditions, we can apply the technique of differential photometry to determine the brightness of our star without ever having to measure A, $\Delta\nu$, and η_ν.

To see how this works, we must become familiar with the stellar magnitude system, where in general, the difference in magnitude between two stars is defined on a logarithmic scale by:

$$m_\nu - {m_\nu}^* = -2.5\ log_{10}\left(\frac{F_\nu}{{F_\nu}^*}\right) \tag{3}$$

where ${m_\nu}^*$ is the magnitude of the standard star, m_ν is the magnitude of the star your are observing, and where ${F_\nu}^*$ and F_ν represent the flux of the standard star and your star, respectively. The reference scale is defined by the star Vega, which is defined as having ${m_\nu}^* = 0$ at all wavelengths. Setting ${m_\nu}^* = 0$ in the above equation, we can write the above equation in terms of the flux of Vega, $F_{\nu 0}$:

$$m_\nu = -2.5\ log_{10}\left(\frac{F_\nu}{F_{\nu 0}}\right) \tag{4}$$

Noting that from Equation 2, $F_\nu \propto (N/t)$ (where N is the total number of photons detected from your star and t is the exposure time), we can take advantage of the properties of logarithms and notice that from equation 3, the difference in magnitude of two stars is given in terms of their flux ratio, which under identical observational conditions allows us to cancel the terms that are difficult to measure, namely A, $\Delta\nu$, and η_ν, leaving everything in terms of N^*, t^*, N, and t. We can see this analytically by substituting equation 2 into equation 3, and after doing some algebra, we arrive at the equations:

$$m_\nu - {m_\nu}^* = -2.5\ log_{10}\left(\frac{Nt^*}{N^*t}\right) \quad \Rightarrow \tag{5}$$

$$m_\nu = m_\nu^* - 2.5\ log_{10}\left(\frac{N}{t}\right) + 2.5\ log_{10}\left(\frac{N^*}{t^*}\right). \tag{6}$$

Using the above equation, having measured N, t, we can determine the magnitude of a star m_ν, provided we also measure a N^*, and, t^* for a standard star of known magnitude ${m_\nu}^*$ under identical observational conditions. Fortunately, many astronomers such as Landolt have done the difficult work of determining the magnitudes of many standard stars using absolute photometry, where they actually measure the numbers we avoided measuring with differential photometry. These standard stars thus provide the foundation for the much easier task of differential photometry.

6.2.2. What is Aperture Photometry?

In the previous discussion, what we did not explain is exactly how to use our CCD camera to measure N and t, which are required for differential photometry. The technique of aperture photometry is one such method that allows us to measure N and t. The CCD is a pixel array and the .fit and .imh images we deal with are basically just two dimensional arrays (say 500 $\times$ 500 pixels), where each pixel contains a measure of the number of photons counted in the given wavelength band over the course of the exposure. Aperture photometry itself involves selecting an aperture to surround the star in the image, usually a circular one, and measuring the counts from the star inside the aperture. This essentially means isolating a set of pixels that forms a circle around your star and counting the total number of counts in all those pixels. Unfortunately, there are other sources of counts measured by your CCD inside the aperture, including dark current, cosmic rays, and sky background counts. We have already discussed how to remove the first two, but the process of aperture photometry also removes the unwanted counts from the sky background.[38]

To measure the sky background, we make an annulus around our aperture called the "sky annulus", far enough from the star that the annulus contains only counts from the sky background, but not too far that it intersects the light from other stars in your field.[39] Basically, we want the sky annulus to contain only sky light, as its name suggests. We then extract the star signal by subtracting the average sky background per pixel in the sky annulus from all the pixels in a small circular aperture containing the star. In the end, we count the number of photons that we believe came from the star alone, to whatever accuracy we can achieve. This is the essence of aperture photometry.

[38]By "sky" here we just mean the molecules in the Earth's atmosphere which absorb and emit light which might ultimately hit our CCD and end up in the aperture where we are trying to measure only the light from our star.

[39]In crowded star fields, where just about any sized sky annulus is bound to intersect light from field stars, you must use a more sophisticated technique than aperture photometry, such as PSF photometry (or other convolution techniques), which we will discuss in future sections.

6.2.3. How to Perform Aperture Photometry Manually

Aperture photometry is sufficient for supernovae that occur far from the nucleus of their host galaxy (no galaxy subtraction needed), and occur in a relatively uncrowded starfield (for which you would need PSF photometry). You can perform aperture photometry in IRAF with the task **daophot**, and its subtask **phot**. In the daophot software package, there is a bunch of image processing software which allows you to construct an aperture and sky annulus and perform aperture photometry. The radius of the aperture (in pixels) and the inner and outer radius of the sky annulus (in pixels) are the kinds of parameters that you input into IRAF. IRAF will sort of act like a black box if you don't understand the code of the programs themselves, but they are essentially doing what was described in the previous section.

If you type daophot on the IRAF command line, it will display a list of the subtasks of the daophot software package as shown below

```
cl>  daophot
da>                                                    # The prompt now becomes da>
     addstar       daotest       nstar      pexamine   psf
     allstar       datapars@     pcalc      pfmerge    psort
     centerpars@   findpars@     pconcat    phot       pstselect
     daoedit       fitskypars@   pconvert   photpars@  seepsf
     daofind       group         pdump      prenumber  setimpars
     daopars@      grpselect     peak       pselect    substar
```

You can type "bye" to get out of daophot without closing IRAF.

In general, in the next few sections, I will discuss how to edit the parameters of all the relevant tasks in IRAF, and display the parameter list as it appears in the xterm when you're in IRAF.[40] In general, I will first display the parameter lists as they appear in IRAF, with the exception that I will highlight the most important parameters in bold face. Afterwards, I will display a list of just those important parameters, and give you some idea of what they are (only if I actually know) and how to determine them. But in general, many of the parameters can be left at their default values.

Basically, you use the IRAF subtask of daophot called phot to perform aperture photometry. phot has several major parameters including image (the image your are performing photometry on), and coords (the coordinates of the supernova and standard stars if you've made a coordinate file, which will be discussed later.) It also contains several major parameters which have sub parameter lists of their own. For example, the parameters: datapar, centerpar, fitskyp, and photpar all have their own separate parameter lists which will be discussed in the following pages.

---

[40] You've already seen this format when I discussed the IRAF task cosmicrays. As such, to keep these parameter lists together for clarity, I may need to introduce a bunch of artificial page breaks that leave a bunch of white space. Please ignore these, and forgive me for the waste of trees.

Parameters that have parentheses may also have a menu of sub parameters which can be edited. If so, they are called "pset" parameters. Here's how to edit the sub parameter list, return to the main list, and eventually run the task manually. A command line without a prompt will open up every time you see an IRAF parameter list, and you can access that command line and execute commands by hitting a colon ":" and typing the keystroke for that command. Here are the most important ones.

:e	# edit the sub parameter list
:q	# quit the sub parameter list, return to main parameter list
:g	# in the main list, this runs the task (g for "go")

Before you begin using IRAF tasks for photometry, you want to pick an image to do photometry on, and display it in the ximtool and get the average Full Width Half Maximum (FWHM)[41] of the stars in the image. You can then edit the parameters of phot, and run the task. But first type:

cl>daophot	# starts daophot, prompt changes to da>
da>disp *imagename*.imh 1	# display the image in the first frame of the ximtool
	# it can be a .fit or a .imh image in general
da>imexam	# opens image examine (imexam) mode

Now a cursor will pop up in the ximtool. Just hit 'a' on 4 or 5 bright standard stars (or the SN), and hit 'q' in the ximtool to quit. Each time it will output a line of text in the xterm that looks something like this, for, say, two stars:

#	COL	LINE	COORD							
#	R	MAG	FLUX	SKY	PEAK	E	PA	ENCLOSED	GAUSSIAN	DIRECT
	314.96	261.04	314.96	261.04						
	9.87	12.92	68104.	172.	4443.	0.07	-86	3.78	**3.43**	3.29
	89.84	99.67	89.84	99.67						
	10.01	14.88	11207.	143.8	695.9	0.08	-70	3.75	**3.47**	3.34

Just pay attention to where it says "GAUSSIAN". This is the FWHM. Just take the average of the numbers in your head that I've emphasized here in bold. For here I'd say the FWHM ≈ 3.45. Most of the time, you don't worry about calculating it exactly since it just needs to be approximate, and in reality, you can't even define a true FWHM for the entire image. But in any case, write down this FWHM, since you'll need it as an input parameter for the various IRAF tasks you're about to do photometry with. Now we can edit the parameters.

[41]The FWHM is a number describing the profile of a symmetric mathematical function such as a Gaussian or bell curve. It is the full width of the curve, as measured at a position half way between 0 and the maximum.

da> epar phot # to edit the parameters of phot. A list will pop up in IRAF as shown below

I R A F
Image Reduction and Analysis Facility

PACKAGE	=	daophot	
TASK	=	phot	
image	=	s1_B0725.imh	Input image(s)
coords	=	sn.cor	Input coordinate list(s) (default: image.coo.?)
output	=	default	Output photometry file(s) (default: image.mag.?)
skyfile	=		Input sky value file(s)
(plotfil	=	)	Output plot metacode file
(datapar	=	)	Data dependent parameters
(centerp	=	)	Centering parameters
(fitskyp	=	)	Sky fitting parameters
(photpar	=	)	Photometry parameters
(interac	=	no)	Interactive mode ?
(radplot	=	no)	Plot the radial profiles?
(verify	=	)_.verify)	Verify critical phot parameters ?
(update	=	)_.update)	Update critical phot parameters ?
(verbose	=	)_.verbose)	Print phot messages ?
(graphic	=	)_.graphics)	Graphics device
(display	=	)_.display)	Display device
(icomman	=	)	Image cursor: [x y wcs] key [cmd]
(gcomman	=	)	Graphics cursor: [x y wcs] key [cmd]
(mode	=	ql)	

image	=	*imagename*.imh	# Input image name (or list of names for automatic photometry later)
coords	=		# No coordinate file yet, we'll use one later for automatic photometry
(datapar	=	)	# Has a list of sub parameters you will also need to edit.
(centerp	=	)	# Has a list of sub parameters you will also need to edit.
(fitskyp	=	)	# Has a list of sub parameters you will also need to edit.
(photpar	=	)	# Has a list of sub parameters you will also need to edit.

Table 11: Here are the most important parameters for phot and how to edit them. The last 4 have sub parameters which we will show how to edit in the following pages. The less important parameters can be left on their default settings, as shown in the upper list in non-bold text.

Now you can edit the sub parameter lists for datapar, centerp, fitskyp, and photpar .

da> :e datapar # to edit the parameters of datapar, sub parameters of phot

I R A F

Image Reduction and Analysis Facility

PACKAGE	=	daophot	
TASK	=	datapars	
(scale	=	1.)	Image scale in units per pixel
(fwhmpsf	=	4.)	FWHM of the PSF in scale units
(emissio	=	yes)	Features are positive ?
(sigma	=	INDEF)	Standard deviation of background in counts
(datamin	=	INDEF)	Minimum good data value
(datamax	=	INDEF)	Maximum good data value
(noise	=	poisson)	Noise model
(ccdread	=	)	CCD readout noise image header keyword
(gain	=	)	CCD gain image header keyword
(readnoi	=	16.)	CCD readout noise in electrons
(epadu	=	5.)	Gain in electrons per count
(exposur	=	EXPTIME)	Exposure time image header keyword
(airmass	=	AIRMASS)	Airmass image header keyword
(filter	=	FILTERS)	Filter image header keyword
(obstime	=	)	Time of observation image header keyword
(itime	=	180.)	Exposure time
(xairmas	=	1.2430000305176)	Airmass
(ifilter	=	V)	Filter
(otime	=	INDEF)	Time of observation
(mode	=	ql)	

(fwhmpsf = 4)	# fwhmpsf = the FWHM, rounded up to the nearest integer # So in our example with a FWHM ≈ 3.45, we get fwhmpsf = 4

Table 12: Here is the most important parameter for datapars and how to edit it. As usual, the less important parameters can be left on their default settings, shown in the upper list in non-bold text. But you should check that they are at the defaults.

```
:q          # to get back to the phot menu
:e centerp  # to edit the parameters of centerpars, sub parameters of phot
```

I R A F
Image Reduction and Analysis Facility

PACKAGE	=	daophot	
TASK	=	centerpars	
(calgori	=	centroid)	Centering algorithm
(cbox	=	7.)	Centering box width in scale units
(cthresh	=	0.)	Centering threshold in sigma above background
(minsnra	=	1.)	Minimum signal-to-noise ratio for centering algo
(cmaxite	=	10)	Maximum iterations for centering algorithm
(maxshif	=	3.)	Maximum center shift in scale units
(clean	=	no)	Symmetry clean before centering
(rclean	=	1.)	Cleaning radius in scale units
(rclip	=	2.)	Clipping radius in scale units
(kclean	=	3.)	K-sigma rejection criterion in skysigma
(mkcente	=	no)	Mark the computed center
(mode	=	ql)	

(calgori = centroid)	# Just make sure the centering algorithm calgori = centroid

Table 13: Here is the most important parameter for centerpars and how to edit it. As usual, the less important parameters can be left on their default settings, shown in the upper list in non-bold text. But you should check that they are at the defaults.

```
:q          # to get back to the phot menu
:e fitskyp  # to edit the parameters of fitskypars, sub parameters of phot
```

I R A F

Image Reduction and Analysis Facility

PACKAGE	=	daophot	
TASK	=	fitskypars	
(salgori	=	mean)	Sky fitting algorithm
(annulus	=	20.)	Inner radius of sky annulus in scale units
(dannulu	=	5.)	Width of sky annulus in scale units
(skyvalu	=	0.)	User sky value
(smaxite	=	10)	Maximum number of sky fitting iterations
(sloclip	=	0.)	Lower clipping factor in percent
(shiclip	=	0.)	Upper clipping factor in percent
(snrejec	=	50)	Maximum number of sky fitting rejection iteratio
(sloreje	=	3.)	Lower K-sigma rejection limit in sky sigma
(shireje	=	3.)	Upper K-sigma rejection limit in sky sigma
(khist	=	3.)	Half width of histogram in sky sigma
(binsize	=	0.10000000149012)	Binsize of histogram in sky sigma
(smooth	=	no)	Boxcar smooth the histogram
(rgrow	=	0.)	Region growing radius in scale units
(mksky	=	no)	Mark sky annuli on the display
(mode	=	ql)	

(salgori	= mean)	# Just make sure the sky fitting algorithm is mean
		# Others include median and mode

Table 14: Here is the most important parameter for fitskyp and how to edit it. As usual, the less important parameters can be left on their default settings, shown in the upper list in non-bold text. But you should check that they are at the defaults.

:q	# to get back to the phot menu
:e photpar	# to edit the parameters of photpars, sub parameters of phot

I R A F

Image Reduction and Analysis Facility

PACKAGE	=	daophot	
TASK	=	photpars	
(weighti	=	constant)	Photometric weighting scheme
(apertur	=	4)	List of aperture radii in scale units
(zmag	=	24.)	Zero point of magnitude scale
(mkapert	=	no)	Draw apertures on the display
(mode	=	ql)	

(apertur = 4)	# Just like fwhmpsf, we want the aperture to be the FWHM # rounded up to the nearest integer, in our case 4

Table 15: Here is the most important parameter for fitskyp and how to edit it. As usual, the less important parameters can be left on their default settings, shown in the upper list in non-bold text. But you should check that they are at the defaults.

Now you are ready to run the task. Make sure the desired image is open in an ximtool and type:

:q	# to get back to the phot menu
:g	# to run phot

Alternatively, you could hit ":q" to get out of phot, and on the daophot command line, type:

da>phot # to run phot at the command line

If you do this, it will ask you to verify a bunch of parameters that were the last ones set. These are remembered by the file .uparm in your iraf_*your name* directory (i.e iraf_andy for me). If you just edited the parameters, and are happy with them, just hit a bunch of carriage returns and phot will run. But in either case, either by hitting :g to go or typing phot at the command line, when phot stars, here's what happens.

First, a cursor will pop up in the ximtool. Without using a coordinate file, you will have to middle click on the supernova to get maximum magnification, then hit 'spacebar' on the center of the supernova. Then you do the same thing, magnify and hit 'spacebar' on the center of all the standard stars, **but you must do so in the order that is indicated by the numbers on your finding chart!** If you do not click on the stars in order, it will be impossible to do accurate photometry. That said, each time you hit 'spacebar' on a star, a line of text will pop up in the corresponding xterm that looks something like:

```
B0725.fit   378.11   296.30   14.18834   19.399   ok
```

These include the x and y coords where you clicked, some preliminary magnitude information, and whether or not everything went ok. When you're done, hit 'q' in the ximtool, and 'q' again in the xterm, and you've successfully done photometry on an image. When finished, phot will output a .mag file into your directory, with a filename that might look like B0725.fit.mag.1 Eventually, we will use these files to extract the magnitude information to construct your instrumental light curves.

Now, for clarity, I will present an abridged version of the commands necessary to run aperture photometry manually on a single image. In IRAF, do the following:

Manual Aperture Photometry (phot) (Quick Reference List)

cl>daophot	# Open the daophot software package
da>disp s1_B0725.imh 1	# Display the relevant image
da>imexam	# Hit 'a' on 4-5 stars to find average FWHM (3.45 for us), 'q' to quit
	#
da>epar phot	# Edit the parameters of phot
image = s1_B0725.imh 1	# Enter the input image for manual photometry
	# Check to make sure the other parameters are set to their default
	# values, as specified in the earlier parameter lists
:e datapar	# Edit the parameters of datapar
fwhmpsf = 4	# FWHM rounded to nearest integer, hit :q to go back to phot
:e centerp	# Edit the parameters of centerp
calgori = centroid	# hit :q to go back to phot parameter list
:e fitskyp	# Edit the parameters of fitskyp
salgori = mean	# hit :q to go back to phot parameter list
:e photpar	# Edit the parameters of photpar
apertur = 4	# also FWHM rounded to nearest integer, hit :q to go back to phot
:g	# to run phot

Now magnify stars and hit spacebar on center in the order specified by the finding chart. When done, hit 'q' in the ximtool and 'q' in the xterm, and phot will output a .mag file to the directory you ran it in. If you mess up, you can always delete the .mag file later. This .mag file contains the instrumental magnitudes for your supernova and standard stars that you will need to construct your instrumental and standard light curves. That's it for manual aperture photometry.

6.3. Point Spread Function (PSF) Photometry

In crowded star fields, where just about any sized sky annulus is bound to intersect light from field stars, you must use a more sophisticated technique than aperture photometry, such as PSF photometry. In this section, we will briefly discuss what PSF photometry is and finally explain how to do it using IRAF.

6.3.1. What is a Point Spread Function?

PSF itself stands for "point spread function", and I will only briefly discuss what it means here. In general, when we are doing photometry, we are interested in measuring the brightness of astronomical point sources such as stars or SNe. A point source is not really a point, but it is effectively so far away that we can not resolve its structure, and instead, all the light from the object appears to be emanating from a single point in space. What this means for CCD observations, is that ideally, all the light from a star or any point source should land in a single pixel! However, from looking at any normal images, we know this is not the case. Starlight is always spread over several pixels, usually in a circular patter, brightest towards the center. Surface plots of the star show a profile that looks like a 3-D Gaussian or bell curve, not a delta function-like spike we would expect for an ideal point source whose light lands on a single pixel. Cosmic rays can often land on just a single pixel, yielding a spiky profile, but stars never do, and in fact, this is a major criteria we have for distinguishing between stars and cosmic rays. So if the star is a point source, why doesn't its light land in a single pixel? What gives?

The answer is that the light, in its journey from the star to our telescope and CCD camera, ultimately has to pass through the Earth's atmosphere. As it passes through the atmosphere, the paths of the photons get bent, and as they deviate from a straight line, they fail to land on a single pixel and are effectively spread or smeared out over several pixels. In fact, if each photon is deflected away from a straight line perpendicular to its motion in random directions, what we expect is for them to land in some circle of pixels about the central point and generate a roughly Gaussian profile, where the number of counts are highest toward the center and then fall off exponentially toward the edges. The mathematical description of how the light from a point source is spread out as to no longer fall on a single pixel is called the "point spread function" or PSF.

If the CCD camera subtends only a small fraction of the sky during the time of the exposure, we can make the rough approximation that the slice of atmosphere in the field of view of the CCD remains nearly constant. This means that all the photons which pass through that slice of atmosphere on the way to the CCD are affected, on average, in the same way as the atmosphere perturbs their paths. What this is saying is that we can approximate that all the stars in the images have the same point spread function. Based on this assumption, we can construct a PSF fit for the image, and use it to do photometry without an aperture. The mathematical details of how this works are beyond the scope of this paper (involving convolution techniques), but it will be useful to recognize that the point spread function is a measure of how the photons are affected by the Earth's atmosphere in their journey from the supernova to our telescope and CCD camera.

6.3.2. *How to Perform PSF Photometry Manually*

PSF fitting photometry is ideal for supernovae that occur in crowded starfield, where the stars are very close together. In all cases, (whether you have done galaxy subtraction or not) you would do aperture photometry first and then do PSF photometry. They must be done in this order. The IRAF package to construct the PSF for your image is also in daophot and is appropriately called **psf**. Following psf, you use another daophot package called **allstar** to actually fit the PSF to the image. Together, they allow you to do PSF fitting photometry. Just as with aperture photometry, we will highlight the most important parameters in bold and discuss how to edit them. Again, you begin by doing it manually on an image, and eventually, in the next section, we will discuss how to do both aperture and PSF fitting photometry. To do PSF fitting manually, first display the first image in the ximtool and type the following as begun on the next page.

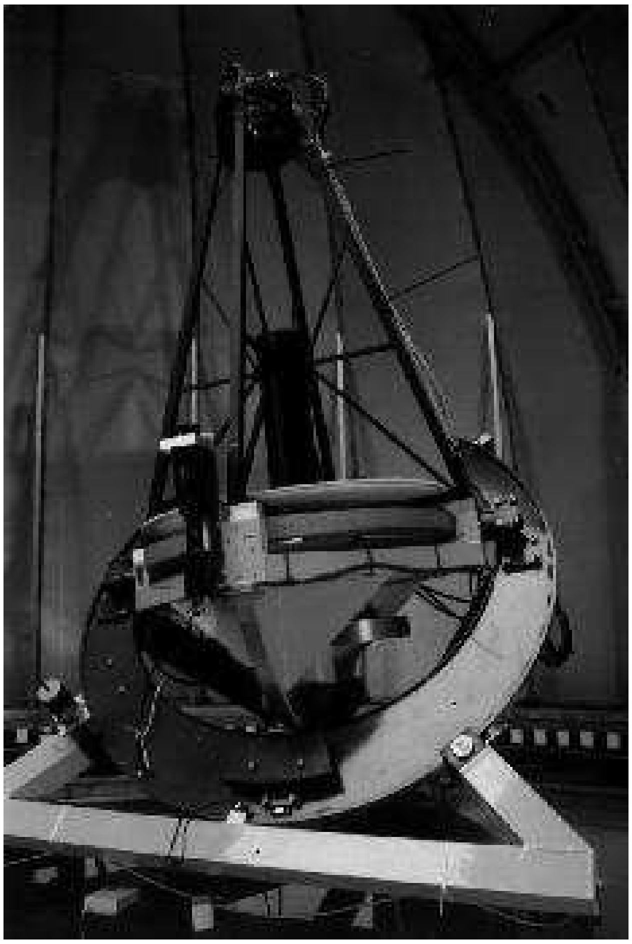

Fig. 15.— Token space filler, (not directly related to PSF photometry): The 76 cm Katzman Automatic Imaging Telescope (KAIT) located at Lick Observatory. The world's most successful low z supernova search telescope, and the ultimate source of nearly all of your photometric data.

```
cl> daophot      # to enter the daophot package
da> epar psf     # to edit the parameters of psf
```

PACKAGE	=	daophot	
TASK	=	psf	
image	=	s1_B0725.imh	Input image(s) for which to build PSF
photfile	=	default	Input photometry file(s) (default: image.mag.?)
pstfile	=		Input psf star list(s) (default: image.pst?)
psfimage	=	default	Output PSF image(s) (default: image.psf.?)
opstfile	=	default	Output PSF star list(s) (default:image.pst.?)
groupfil	=	default	Output PSF star group file(s) (default: image.ps
(plotfil	=	)	Output plot metacode file
(datapar	=	)	Data dependent parameters
(daopars	=	)	Psf fitting parameters
(matchby	=	no)	Match psf star list to photometry file(s) by id
(interac	=	yes)	Compute the psf interactively ?
(showplo	=	yes)	Show plots of PSF stars?
(plottyp	=	mesh)	Default plot type (mesh—contour—radial)
(verbose	=	)_.verbose)	Print psf messages ?
(verify	=	)_.verify)	Verify critical psf parameters ?
(update	=	)_.update)	Update critical psf parameters ?
(graphic	=	)_.graphics)	Graphics device
(display	=	)_.display)	Display device
(icomman	=	)	Image cursor: [x y wcs] key [cmd]
(gcomman	=	)	Graphics cursor: [x y wcs] key [cmd]
(mode	=	ql)	

image	=	s1_B0725.imh	# Input image to do PSF manually
pstfile	=		# None now, but you make and use one later for automatic PSF photometry
(datapar	=	)	# Same parameters were used for phot, so you can likely leave datapar alone
(daopars	=	)	# These have a sub parameter list which must be edited

Table 16: The most important parameters for psf and how to edit them. As usual, for the less important parameters, just check that they're at the default settings, shown above in non-bold text.

:q	# to get back to the psf menu
:e datapars	# to edit the parameters of datpars, sub parameters of psf (and phot) if you
	# even need to. The same parameters were used for aperture photometry
	# But just in case they've been edited since, we'll display them again here.

I R A F
Image Reduction and Analysis Facility

PACKAGE	=	daophot	
TASK	=	datapars	
(scale	=	1.)	Image scale in units per pixel
(fwhmpsf	=	4.)	FWHM of the PSF in scale units
(emissio	=	yes)	Features are positive ?
(sigma	=	INDEF)	Standard deviation of background in counts
(datamin	=	INDEF)	Minimum good data value
(datamax	=	INDEF)	Maximum good data value
(noise	=	poisson)	Noise model
(ccdread	=	)	CCD readout noise image header keyword
(gain	=	)	CCD gain image header keyword
(readnoi	=	16.)	CCD readout noise in electrons
(epadu	=	5.)	Gain in electrons per count
(exposur	=	EXPTIME)	Exposure time image header keyword
(airmass	=	AIRMASS)	Airmass image header keyword
(filter	=	FILTERS)	Filter image header keyword
(obstime	=	)	Time of observation image header keyword
(itime	=	180.)	Exposure time
(xairmas	=	1.2430000305176)	Airmass
(ifilter	=	V)	Filter
(otime	=	INDEF)	Time of observation
(mode	=	ql)	

(fwhmpsf	= 4)	# fwhmpsf = the FWHM, rounded up to the nearest integer
		# So in our example with a FWHM ≈ 3.45, we get fwhmpsf = 4

Table 17: Here is the most important parameter for datapars and how to edit it. As usual, the less important parameters can be left on their default settings, shown in the upper list in non-bold text. But you should check that they are at the defaults.

```
:q            # to get back to the psf menu
:e daopars    # to edit the parameters of daopars, sub parameters of psf
```

PACKAGE	=	daophot	
TASK	=	daopars	
(functio	=	gauss)	Form of analytic component of psf model
(varorde	=	0)	Order of empirical component of psf model
(nclean	=	0)	Number of cleaning iterations for computing psf
(saturat	=	no)	Use wings of saturated stars in psf model comput
(matchra	=	3.)	Object matching radius in scale units
(psfrad	=	20.)	Radius of psf model in scale units
(fitrad	=	3.75)	Fitting radius in scale units
(recente	=	yes)	Recenter stars during fit ?
(fitsky	=	no)	Recompute group sky value during fit ?
(groupsk	=	yes)	Use group rather than individual sky values ?
(sannulu	=	0.)	Inner radius of sky fitting annulus in scale uni
(wsannul	=	20.)	Width of sky fitting annulus in scale units
(flaterr	=	0.75)	Flat field error in percent
(proferr	=	5.)	Profile error in percent
(maxiter	=	50)	Maximum number of fitting iterations
(clipexp	=	6)	Bad data clipping exponent
(clipran	=	2.5)	Bad data clipping range in sigma
(mergera	=	INDEF)	Critical object merging radius in scale units
(critsnr	=	1.)	Critical S/N ratio for group membership
(maxnsta	=	10000)	Maximum number of stars to fit
(maxgrou	=	60)	Maximum number of stars to fit per group
(mode	=	ql)	

(functio	=	gauss)	# Assume Gaussian profile for stars
(psfrad	=	20.)	# Set radius of PSF model = 20
(fitrad	=	3.45)	# fitrad = FWHM (3.45 in our earlier example)
(wsannul	=	20.)	# wsannul = fitrad = 20 (both should be the same)

Table 18: The most important parameters for psf and how to edit them. As usual, for the less important parameters, just check that they're at the default settings, shown above in non-bold text.

Now you're ready to run psf. Type:

:q	# to get back to the psf menu
:g	# to run psf (you may need to hit a few carriage returns to verify parameters)

Now a cursor will pop up in your image, and you need to select your PSF stars, the ones that will be used to construct a PSF fit for the image. In general, you want to choose the brightest standard stars, and in almost all case, **do not use the supernova as a PSF star!**[42] Just use, say the 4 or 5 brightest standard stars that are well separated from one another. If there are many bright stars in the images, you can use more, but don't exceed 9 or 10 PSF stars. If the field is relatively barren, and has maybe only 2 stars, you still should use what you have. The PSF fit will be less accurate, but its the best we can do under the circumstances.

To select PSF stars, hit 'a' on the desired star in the ximtool. A surface plot (3D profile) of the star will then pop up in a TeK window. If you want to "un-choose" the star you just selected, hit 'd' in the TeK window and a line of text will pop up that says "Star X rejected by user.", where X is whatever the number of star you clicked is on the finding chart.[43] To finalize your choice of the PSF star, hit 'a' in the TeK window and a happy output will pop up in the xterm that lists the star number and some info about the star you just used to help construct the PSF for the image.[44] When you've selected all your psf stars, hit 'q' in the ximtool and hit 'q' again in the xterm. Afterwards, if everything worked it will say "constructed PSF for image".

It will output several files to the directory in which it was run with names that look something like this. The file extensions are what is important to look at.

s1_B0725.imh.psf.1.imh	# The psf image
s1_B0725.imh.psg.1	# PSF group file, you only deal with this for very crowded fields
s1_B0725.imh.pst.1	# PSF star list. This is very important. You use this as input for
	# automatic PSF photometry, to avoid choosing PSF stars every time.

But you haven't finished PSF photometry yet. You need to run one more task, called allstar. The task psf created a PSF for your image, but the allstar task actually fits this model PSF to the image itself. You need to compare the fit to the model for PSF fitting photometry.

[42]In rare cases where the SNe is very far from the host galaxy, where there aren't enough good standard stars, it can be used as a PSF star, but if you have enough other good PSF stars, do not use it. And certainly do not use it if it is close to the galactic nucleus and you've already performed galaxy subtraction.

[43]Since you've already done aperture photometry and generated a .mag file where the supernova and standard stars are numbered, this file is input into the psf package as the parameter photfile, and thus psf knows the numbers of the stars that correspond to your finding chart.

[44]I discovered that you can hit 'w' in the Tek window to rotate the surface plot and make sure that a PSF star is not too close to another star and not hit by an un-removed cosmic ray.

da>epar allstar # to edit the parameters of allstar

PACKAGE	=	daophot	
TASK	=	allstar	
image	=	s1_B0725.imh	Image corresponding to photometry
photfile	=	default	Input photometry file (default: image.mag.?)
psfimage	=	default	PSF image (default: image.psf.?)
allstarf	=	default	Output photometry file (default: image.als.?)
rejfile	=	default	Output rejections file (default: image.arj.?)
subimage	=	default	Subtracted image (default: image.sub.?)
(cache	=	yes)	Cache the data in memory?
(datapar	=	)	Data dependent parameters
(daopars	=	)	Psf fitting parameters
(verbose	=	).*verbose*)	Print allstar messages?
(verify	=	).*verify*)	Verify critical allstar parameters?
(update	=	).*update*)	Update critical allstar parameters?
(mode	=	ql)	

image	=	s1_B0725.imh	# Same image used for phot and psf

Table 19: Set the non-bold parameters to their defaults. allstar has the sub parameter lists for datapar and daopars, but if you've just run phot and/or psf, you shouldn't need to modify them.

:g # to run allstar

When finished, allstar will output these files into the directory where you ran it.

s1_B0725.imh.als.1	# This file has the allstar/psf photometry info (like the .mag file for phot)
s1_B0725.imh.arj.1	# I don't know what this is, but its not important...ask Weidong.
s1_B0725.imh.sub.1.imh	# The subtracted image, with PSF stars missing. You fit a PSF to the image,
	# and subtract it from the image, and a measure of how well the model fit the
	# actual image by seeing how much residual starlight remains in the subtraction.

The .als file is the analogue of the .mag file that was generated for aperture photometry. It also contains the instrumental magnitudes of your supernova and standard stars, obtained by the alternative method of PSF fitting photometry. It will also be needed to generate your instrumental and standard light curves.

Now, for clarity, I will present an abridged version of the commands necessary to run psf and allstar photometry manually on a single image. In IRAF, do the following:

Manual PSF Photometry (psf and allstar) (Quick Reference List)

cl>daophot	# If you just ran phot, you're already in daophot
da>disp s1_B0725.imh 1	# Display the relevant image
da>epar psf	# Edit the parameters of psf
image = s1_B0725.imh	# Enter the input image for manual psf photometry
	# Check to make sure the other parameters are set to their defaults
:e datapar	# Edit the parameters of datapar
fwhmpsf = 4	# FWHM rounded to nearest integer, hit :q to go back to phot
	# If you've just run phot, you don't need to edit these again
:e daopars	# Edit the parameters of daopars
functio = gauss	# PSF fit model for star profiles
psfrad = 20	# Radius of the PSF model, set psfrad = 20
fitrad = 3.75	# fitrad = FWHM
wsannul = 20	# wsannul = psfrad = 20
:q	# hit :q to go back to psf parameter list
:g	# to run psf

'a'	# hit 'a' on the desired star in the ximtool to select PSF stars.
	# A surface plot (3D profile) of the star will then pop up in a TeK window.
'd'	# To "un-choose" the star you just selected, hit 'd' in the TeK window
	# a line of text will pop up that says "Star X rejected by user."
'a'	# Hit 'a' again in the TeK window to finalize your choice of the PSF star.
	# a happy output will pop up in the xterm

When you've selected all your psf stars, hit 'q' in the ximtool and hit 'q' again in the xterm. Afterwards, if everything worked it will say "constructed PSF for image". psf will output files with extensions like .psf.1.imh, .psg, and .pst into the directory you ran it in. Now run allstar.

da>epar allstar	# Edit the parameters of allstar
image = s1_B0725.imh	# Enter the input image for manual allstar photometry
:q	# hit :q to go back to allstar parameter list
:g	# to run allstar, then just hit a bunch of carriage returns

This should output an .als file into the directory you ran psf and allstar in. It also outputs .arj and .sub.1.imh files but the .als file is the important one, since it has the instrumental magnitudes for your supernova and standard stars in that image. That's the story.

6.4. How to Perform Aperture and PSF Photometry Automatically

You've just seen how to run phot, psf, and allstar to do aperture and PSF photometry on individual images manually. Now, we can discuss how to use Weidong's program autophot.cl to do exactly these same steps automatically. Doing the steps manually on one or two images is necessary since it creates a few required files for the automatic program, and it sets many of the parameters to default values that will be used again throughout the automatic program. It is also a good idea because it ultimately lets you know what the automatic program is doing when you actually use it. In the end, the automatic program drastically speeds up the photometry process and give you a sense and a respect for the power of computation, after seeing how long it takes to be the computer yourself.

*6.4.1. Registering the Images**

Before we can do aperture and PSF photometry automatically, we need to register the images. Registering the images means aligning them so that the supernova and all the standard stars all have the same x-y coordinates in all the images. Having the images registered allows you to do photometry automatically on your images because now the program will know where the supernova and standard stars are in all of the images. You won't have to individually magnify and click on the supernova and standard stars in all of the images, which is a significant pain in the ass. First off, if you've already done galaxy subtraction, you can skip this step because the galaxy subtraction programs already register the images for you. Hence the asterisk* in the title of this section. Otherwise, if you are not doing galaxy subtraction, you can register the images using one of Weidong's programs called **register.cl**[45]

First, you have to create a template image to register around, (as you might have seen with galaxy subtraction.) To create a template, first look through your images and select and image from any band that contains the supernova and all the standards from your finding chart, and generally is of good quality. You can construct a template from this image using the IRAF task **automktem_t**. First, you must be in daophot. Then type:

[45]You could presumably learn how to write a program yourself in IDL that would register the images but this might be getting a bit more advanced for now.

da> epar automktem_t # edit the parameters for the IRAF task **automktem_t**,
which is its standard template making program.

I R A F
Image Reduction and Analysis Facility

PACKAGE	=	user	
TASK	=	automktem_t	
inputA	=	B98dh.imh	image name
fieldnam	=	B98dh	name of field
delnot	=	no	Delete fit after mktemp?
(three	=	no)	Combine 3 files
(tmin	=	10.)	min thresh for dophot
(tmax	=	10000.)	max thresh for dophot
(FWHM	=	2.83)	Approximate FWHM of images
(saturat	=	50000.)	level for saturation
(oblit	=	4.)	obliteration size
(smooth	=	no)	Do you want to smooth image
(smoothv	=	1.5)	amount to smooth (FWHM)
(dof	=	yes)	Is this for findsn?
(fwfile	=	/cetus/weidong/temp/ssigma.dat)	
(shfile	=	sexshift.dat)	
(galfile	=	/whale/weidong/template/F20918.gal)	
(mode	=	ql)	

inputA	=	B98dh.imh	# input image name used to create template
fieldnam	=	B98dh	# name of field, usually filter and supernova is a good idea

Table 20: Again, here are the most important parameters and how to edit them.

When you're ready to run the task, type:

:g # to run automktem_t

automktem_t will spit out these kinds of files.

B98dh.fit	# The actual template image
B98df.fwhm	# Contains information you would get by hitting 'a' on
	# stars in imexam mode (i.e GAUSSIAN)
B98dh.info	# contains important information about the template like
	# star coordinates but you don't need to worry about it now
B98df.stars	# I don't really know what this is...ask Weidong

These files should be familiar to you if you've already done galaxy subtraction.

Now, once you've made the template, you can move on toward registering the images using Weidong's program register.cl. You need to create a list of images to register. This can be done with the "ls *.fit > Blist" command, and then editing Blist to keep only the desired filenames. You'll need to get a copy of register.cl from Weidong.[46] You'll have to make a copy of it and edit it in the directory in which you want to register the images. The code (with comments) looks something like the example on the next page.

The only things in the code you will need to change before running the program are:

list = "flist"	# or whatever you named the list of images to register (i.e Blist)
v02ap	# anywhere it says v02ap, change it to your new template name (i.e B98dh)

Once modified, you can run register.cl with the following commands in IRAF.

cl> task $register=register.cl	# defines register.cl as an IRAF task to be run by typing "register"
cl> register	# This will run the program and register the images

In the end, the program outputs an extra 's' to the beginning of the filenames of the images that have been registered. It will not overwrite the original images. Once you've looked at the registered images to make sure they're all aligned, you can delete the old images and rename the new ones as you desire. Renaming a whole directory of files, such as getting rid of all the prefix s's requires some vi and command line tricks, which are described in the appendix, but you might just want to leave the new filenames as they are. In general, how you keep track of and change your filenames is up to you. The ideas mentioned in here are just suggestions that I have found helpful, but in general, most file naming schemes come down to personal preference, (although some are better than others!).

[46] When you edit your copy of register.cl and look at the code, you'll see a reference to a program named Wqreg, which is the actual image registration program which is called by register.cl

```
struct *shiftf

printf("\n\n Register.cl \n\n")
printf("\n  To use register.cl, try the following steps:")
printf("\n  1) make a template using automktem_t. Select an ")
printf("\n     image that represents most pointing as the template.")
printf("\n  2) modify this register.cl if necessary. Places to watch")
printf("\n     for are preceded by a ### sign.")
printf("\n  3) define the task: cl> task $register=register.cl")
printf("\n  4) run the task: cl> register")
printf("\n\n")
printf("\n  Please wait for 10 seconds ......\n")

### "flist" is the list of image names.

list="flist"

while(fscan(list,s1)!=EOF)
{

### "v02ap" is the template name.

   print("cp "//s1//" v02ap_001.fit") | cl

### "v02ap" is the template name.

   Wqreg("v02ap", threshmi=10., threshma=50000.)

   shiftf="sexshift.dat"
   i=fscan(shift, x, y)
   if(access("s"//s1)) delete("s"//s1)

### "v02ap" is the template name.

   imshift("v02ap_001.fit", "s"//s1, xshift=x, yshift=y)
}
```

Table 21: Example code for IRAF script **register.cl** As mentioned on the previous page, the only things you'll need to change in the program are: list = "flist" (# change to Blist, or whatever you named the list of images to register), v02ap (# anywhere it says v02ap, change it to your new template name, which was B98dh in our example.)

6.4.2. Creating a Coordinate File

A coordinate file is a file with a .cor extension which contains a list of the x and y coordinates for selected objects in the image, which for us means the supernova and the standard stars. It also contains a number for the supernova '1', and for each of the standard stars "2,3,4...". It is crucial that this numbering scheme be the same as indicated on the finding chart. In fact, creating a coordinate file, and using the tvmark command is often the best way to create a finding chart, and you could presumably use the following techniques to create a finding chart for future supernovae that you work on. But for now, let's figure out how to create a coordinate file. First, select a good image that has the supernova and all the standard stars (basically any image, if they've already been registered), and display it in the ximtool. Then type:

```
cl> rimcursor > sn.cor      # Read image cursor (rimcursor), and output the results
                            # to the coordinate file sn.cor
```

Now in the ximtool, magnify on the supernovae and standard stars in the same order as in the finding chart, and at highest magnification for each one, hit spacebar on the center of the star. This will specify its x-y coordinates. Once you've specified the coordinates of the supernova and the standard stars in order, hit "Ctrl d" to quit.[47]

Now, edit the coordinate file, the first few lines will first look something like:

```
81.359 381.783 101 \040     #
68.297 383.858 101 \040     #
113.464 341.377 101 \040    #
```

Now replace the "101" with the finding chart number of the object, corresponding to the order in which you selected the object. Now the coordinate file should look like:[48]

```
81.359 381.783 1 \040     #
68.297 383.858 2 \040     #
113.464 341.377 3 \040    #
```

Now from here, you can actually use the tvmark command to create and print out a finding chart. Presumably, Weidong has given you one for your first supernova, but this technique will allow you to make finding charts for future supernovae if you so desire, even before you register the images. First display the image you made the coordinate file from in, say, frame 1 of the ximtool.

[47]Don't hit 'q' to quit, rimcursor will mess up.

[48]I have no idea what the "\040" means, but it seems harmless, so leave it alone.

Then type:

```
cl> tvmark 1 sn.cor    # The supernova and standard stars are now circled in the ximtool
                       # and marked by numbers, as specified by the coordinate file.
```

Save this image if you wish, and print it out from the ximtool menu and you've got yourself a finding chart. Otherwise, you've also got yourself a coordinate file, which makes things a hell of a lot easier for automatic aperture photometry especially because otherwise, you'd spend hours magnifying and clicking on the supernova and standard stars in order for each of the images you wish to do photometry on. I hope I don't need to convince you that, beyond trying it manually for a few test images, that is a big waste of your time.

6.4.3. Running autophot.cl

The steps you just performed, registering the images and creating a coordinate file, (and the previously finished creation of a PSF star list) all make it possible to do aperture and PSF photometry automatically. autophot.cl is a simple IRAF script. When opened with emacs or vi, the code for which will look something like the example on the next page. Again, it just mimics what you would have typed into the IRAF command line when you did everything manually. Just as with register.cl, to run autophot.cl, you'll need to put a copy of autophot.cl in the directory you wish to do aperture and PSF photometry in, and change a few things in the code.

```
list ="s1_Blist"                  # change to your input list of images to perform photometry on
coords = "sn.cor"                 # change to the coordinate file you created
pstfile="s1_B0725.imh.pst.1"      # change to the PSF star list you created when doing PSF photometry
```

Again, as with tempsub.cl, and register.cl, you must make autophot.cl and IRAF task with:

```
cl> task $autophot=autophot.cl    # defines autophot.cl as an IRAF task to run by typing "autophot"
cl> autophot                      # This will run the program and register the images
                                  # The same coordinate file and PSF star list should work for all
                                  # bands, as long as the registering process used the same template
```

The program will run and all you need to do is enter the average FWHM for the stars in each of the images. In general, the image from the input list will pop up in the ximtool in imexam mode, and you need to hit 'a' on 4-5 of the brightest standard stars in the image.[49] Each time you hit 'a' a line will come up in the xterm, which includes the FWHM. Do the average in your head to figure out the average FWHM for those 4-5 stars. (In general, a FWHM of 3 is normal, and 5 is high and usually indicative of bad weather) Hit 'q' to get out of imexam mode. The program will

[49] Its OK to find the FWHM of the supernova as one of the 4-5 stars if you want.

prompt you for the FWHM in the xterm, Enter it, and then hit a bunch of carriage returns. (16, I believe). After this, you hit 'q' in the ximtool and it will move on to the next image, where you will repeat the process. It doesn't matter what order you click on the 4-5 standard stars, since you're just calculating the average of their FWHM's, but you do need to use the same 4-5 stars each time.

If you ever mess up and forget to enter the FWHM or hit enter too many times, just go back to the input list and delete the names of the images you've already done, save the changes, and then just type "autophot" again on the command line

```
cl> autophot      # This will run the program again using the modified input list
```

When you've finished running autophot.cl in each of the bands, your directory will be filled with output files of the form .mag (aperture photometry), and .als (PSF fitting photometry). These contain the data you will eventually use to construct your instrumental and standard supernova light curves.

```
list ="s1_Blist"
while(fscan(list,s1)!=EOF)
{
displ(s1,2)
imexam
printf("\n FWHM =")
i=scan(x)
j=int(x)+1

  phot(s1, coords="sn.cor", output="default", skyfile=, aper=j, inter-)

  psf(s1, photfile="default", fitrad=x, pstfile="s1_B0725.imh.pst.1",
    psfimage="default", opstfile="default", groupfile="default",verify-, inter-)

  allstar(s1, photfile="default", fitrad=x, psfimage="default",
    allstarf="default", rejfile="default", subimage="default")

imexam(s1//".sub.1")
}
```

Table 22: Example code for **autophot.cl** As mentioned on the previous page, the only things you'll need to change in the program are: list ="s1_Blist" (# or whatever your input list of images to perform photometry on is), coords = "sn.cor" (# make sure you use the coordinate file you created), pstfile="s1_B0725.imh.pst.1" (# make sure you use PSF star list you created when doing PSF photometry).

7. How to Display Your Data Meaningfully: Generating Instrumental and Standard Supernova Light Curves

For all intents and purposes, you are now done with photometry. All the photometric data is now stored in your collection of .mag files (if you did only aperture photometry), and your .als files (if you also did PSF photometry). The remaining task is then to extract the relevant data from these files, and display it in meaningful form. The first step is to sort the data, and then use it construct your instrumental light curves, which plot the supernova's instrumental brightness in magnitudes vs. time in Julian days. This step is not so complicated. The second step is to take the instrumental magnitudes and use a set of transformation equations to change the instrumental magnitudes to a standard scale that all astronomers can agree upon. This step is considerably more complicated, but when you finish, you can generate a plot of your standard light curves in all the bands (magnitude vs. Julian days), and also display the photometric data with errors in a reference table. Example plots of standard light curves and photometric data tables were shown near the very beginning of this primer. After you generate them now, you will have come full circle and you will be completely finished, having reduced the data and displayed it in publishable form.

Now a lot of these steps do require some programming to be done, for example, using IDL. But the truth of it is, the amount of programming you want to learn is up to you, and at this point, you could conceivably just hand over the .mag and .als files to Weidong (along with the lists of Julian dates in each band), and he already has software to sort the data, apply the transformation equations and generate standard light curves. I had hoped to actually discuss how to write some of these programs in IDL, and show some example programs that I wrote, but it has turned out to be beyond the scope of this primer. But I will try to discuss some of the relevant steps and I encourage you to learn how to write a few basic programs which will allow you to sort the data and plot the light curves. So the following sections will discuss what you need to do, but not how to explicitly write code for it.

7.1. Sorting Your Photometric Data: The IRAF txdump Command

The .mag and .als files have a lot of information in them, only some of which is relevant. If you typed "more *filename*.mag.1", you would see a jumble of text displayed. To extract only the relevant text, you can use the IRAF txdump command. For example, go to the B directory with your .mag files, make sure you're in IRAF and type:

```
cl> txdump *.mag im,id,mag,merr yes | sort > Bmag.tab   # Extracts a list of relevant .mag data
                                                        # and puts it into a file named Bmag.tab
cl> txdump *.als im,id,mag,merr yes | sort > Bals.tab   # txdump also works for .als files
                                                        # Extracts a list of relevant .als data
                                                        # and puts it into a file named Bals.tab
```

Bmag.tab and Bals.tab will contain a list of image names (im), identification numbers of the supernova and standard stars (id), the instrumental magnitude of the supernova and standard stars (mag), and the associated magnitude error bars (merr). They will be arranged alphanumerically as a list of rows. The columns will be arranged in exact correspondence to the order you specified in the txdump command [im,id,mag,merr]. For example, excluding the first line, the beginning of Bmag.tab might look like this:

im	id	mag	merr
s1_B0725.imh	1	19.483	0.007
s1_B0725.imh	2	21.293	0.030
s1_B0725.imh	3	20.368	0.010
s1_B0726.imh	1	19.148	0.012
s1_B0726.imh	2	21.483	0.021
s1_B0726.imh	3	20.295	0.011
s1_B0728.imh	1	18.655	0.009
s1_B0728.imh	2	21.374	0.017
s1_B0728.imh	3	20.186	0.008

This is what the list would look like for the first three images, if all images had only the supernova and 2 standard stars. Note that according to the numbering scheme from our finding chart, the supernova is numbered 1, and the other standard stars are numbered 2 and 3 in order.

The yes | sort command sorts the rows into alphanumeric order, so one thing to be wary of is that if you happen to have more than 9 stars (including the SN), the sort command will output a file that, for the first image, looks like this.

s1_B0725.imh	1	19.483	0.007
s1_B0725.imh	10	20.395	0.011
s1_B0725.imh	11	21.155	0.024
s1_B0725.imh	2	21.293	0.030
s1_B0725.imh	3	20.368	0.010
s1_B0725.imh	4	21.160	0.021
s1_B0725.imh	5	22.107	0.031
s1_B0725.imh	6	17.932	0.004
s1_B0725.imh	7	19.867	0.011
s1_B0725.imh	8	22.463	0.034
s1_B0725.imh	9	20.378	0.014

This can get you into trouble when you are programming, because often you are dealing with arrays of numbers where you need to make sure the array indices match up, and the list {1,2,3,4,5,6,7,8,9,10,11} is definitely different from the list {1,10,11,2,3,4,5,6,7,8,9}, so you need to be careful.

If you actually do write a program to sort the data, what you want to do is have it read the contents of the file Bmag.tab (and/or Bals.tab), and sort the data into bins or slots in an array, so that for each day, each standard star has a value of im, id, mag, and merr. If there are, for example, 11 standard stars and 40 days of observation, in the B band, (plus the 4 items: im, id, mag, merr), this would correspond to an (11 x 40 x 4) 3-dimensional array, for a single band. If you wanted to put all of the information for all of the bands (for example 5 bands: U, B, V, R, and I) into a single array, you could create a (5 x 11 x 40 x 4) 4-dimensional array, but that's up to you.[50]

7.2. Generating Your Relative Instrumental Light Curves

From the txdump data, you construct your relative instrumental magnitude in each band for each day of observation by taking the txdump instrumental magnitude of the supernova (id = 1), and subtracting the average txdump instrumental magnitude of each of the standard stars. If you have only a few standard stars, use them all in the average. If you have a large number of standard stars, use only the 4 - 6 brightest ones.[51] In general, if you are going to use all N standard stars with txdump instrumental magnitudes m_i, $i = 2, 3, 4...N$, (where the supernova always has txdump instrumental magnitude m_1, you can calculate the relative instrumental magnitude m_{SnRel} from:

$$m_{SnRel} = m_1 - \left(\frac{1}{N-1}\right)\sum_{i=2}^{N} m_i \tag{7}$$

Or, more explicitly, if, for example, you only used 4 bright standards with id numbers 3, 5, 7, and 9, then m_{SnRel} is given by:

$$m_{SnRel} = m_1 - \left(\frac{m_3 + m_5 + m_7 + m_9}{4}\right) \tag{8}$$

Taking the txdump magnitude of the supernova and subtracting the average of the txdump magnitudes of the brightest standard stars can give you a relative, instrumental light curve because it accounts for the fact that even though the apparent brightness of all the stars (and the SN) may fluctuate from observation to observation due to the weather, in theory, the intrinsic brightness of all the standard stars should remain essentially constant (if they aren't variable stars), while the intrinsic brightness of the supernova should vary in time. By subtracting the average apparent brightness of the standard stars from the apparent supernova brightness, we can deduce the true

[50]IDL uses arrays to store data and refers to the location of the first element in the array a with array index 0, so the first element of array a is a[0], and the second element is a[1]. This is something useful to keep in mind as it can often be confusing. If you know you are going to be plotting the light curves in IDL, it might even be useful to set up your finding chart from the beginning with the supernova as star number 0.

[51]This will reduce the scatter in your light curve because the instrumental magnitude measurement errors will be smaller for the brightest standard stars since they have a higher signal to noise ratio than the fainter stars, in the sense that their photons stand out more from the photons due to the background noise in the image.

shape of the instrumental supernova light curve as it varies in time, roughly independent of the brightness fluctuation from image to image.[52]

Actually writing a program to code this calculation using IDL, for example, is not too bad. First, you want to sort your txdump data into arrays, and then you can do simple arithmetic on the arrays. But for now, its reasonable just to have an idea of how you would construct an instrumental magnitude for the supernova in each of the bands, getting a data point for each day of observation. Once you have those data points also stored into an array, you would plot the contents of that array vs. the contents of an array that has the corresponding Julian dates for each image, and you would then have your instrumental light curve. So you can either learn to do this on your own, or with help from someone, or you could just hand the txdump files over to Weidong. Either way, its up to you.

7.3. Generating Standard Light Curves

Generating relative instrumental light curves is just a quick way to get an idea for the relative shapes of the light curves if we don't have calibrating images which give the standard magnitudes of the standard stars. But a relative instrumental light curve depends on the particular details of the instrument you used to make the observation. These include the optics of your telescope, the characteristics of the color filters for each band, the properties of your CCD camera, and that sort of thing. This will result in a light curve that is unique to your instrument, and thus not verifiable by other astronomers using different instruments. However, it is possible to put the supernova magnitudes onto an observer independent scale that every astronomer can agree upon, and this is what we will discuss here.

The data you have from the txdump command is a list of instrumental magnitudes for the supernova and the standard stars. If we could actually get a list of observer independent *standard* magnitudes for all of our standard stars, we could use them to calibrate the supernova's instrumental magnitude, and find the standard magnitude for the supernova. In fact, we can do this by using the technique of differential photometry to find the standard magnitudes of all the standard stars in your field. When we discussed the difference between differential and absolute photometry earlier, we were preparing ourselves for the technique we will discuss now to generate standard light curves.

What we do is wait for a clear night, use a catalog of Landolt 9or other) standard stars, for which absolute photometry has already been done, and make consecutive observations of the Landolt standards followed by the standard stars in our image. We say that the Landolt standards are the **primary** standards and that the standard stars in our field are our **secondary** standards. We then use differential photometry to find the standard magnitudes of the secondary standard stars in our image, and from there we can do a final extra step of differential to find the standard

[52]However, the photometric errors will certainly be greater on nights with bad weather, since the supernova and standard stars all appear dimmer than they would on a clear night, and thus have a lower signal to noise ratio and higher error bars.

magnitude for our supernova. In a sense, we can view the supernova as our **tertiary** star, since its magnitude is derived from a two step process built up from the Landolt standards and the secondary standards. We can't just go straight from the Landolt standards to the supernova since the supernova intrinsically changes in brightness, while the secondary standards do not intrinsically change in brightness. Once we have a list of the standard magnitudes of our secondary standard stars, which you must get from Weidong, we can combine this with our instrumental magnitudes for our secondary standards and our supernovause and use equations based on differential photometry to transform to standard magnitudes for the supernova. This is achieved via the transformation equations discussed in the next section.

7.4. Transforming to Standard Magnitudes: The Transformation Equations

To construct the equations to transform from instrumental to standard magnitudes, we will use the following notation.[53] An upper case X stands for the standard X band magnitude, while a lower case x stands for the instrumental X band magnitude, where the supernova and standards are denoted with the subscripts X_{sn} and X_s, respectively. X can be either U, B, V, R, or I. When a quantity depends both on time t, and the particular standard star i ($i = 1, 2, ...N$ when there are $N-1$ standard stars), then the quantity will be denoted by $X_s(t,i)$, for example. Quantities that are only time dependent will just look like $x_{sn}(t)$, while time independent quantities that only depend on the particular standard star i look like $X_s(i)$, for example. So, for example, in the B band ($X = B$), the standard magnitude of the supernova as a function of time, as calculated from standard star i is given by:

$$B_{sn}(t,i) = B_s(i) + b_{sn}(t) - b_s(t,i) + b_1\Big[\{\ b_{sn}(t) - v_{sn}(t)\ \} - \{\ b_s(t,i) - v_s(t,i)\ \}\Big] \qquad (9)$$

where the terms are defined explicitly as follows:[54]

$B_{sn}(t,i)$	# The B-band standard magnitude of the supernova at time t, calculated from standard star i
$B_s(i)$	# The B-band standard magnitude of secondary standard star i (does not depend on time)
$b_{sn}(t)$	# The B-band instrumental magnitude of the supernova at time t
$b_s(t,i)$	# The B-band instrumental magnitude of secondary standard star i at time t
b_1	# The B-band instrumental color term, which depends on the properties of the B band filter used

This equation gives you the standard B band magnitude of the supernova at time t, as calculated using standard star i, or $B_{sn}(t,i)$. Now i is actually a subscript that runs from $i = 1, 2, ...N$,

[53] These are all apparent magnitudes, as seen from earth, not intrinsic luminosities, as seen from $10pc$ away.

[54] The color term and its multiplicative factor (a difference between B and V band instrumental magnitudes for the standard stars and the SN) is required to remove instrumental affects. The color terms are defined with respect to the V band, which is why the instrumental V band magnitudes $v_{sn}(t)$ and $v_s(t,i)$ show up in the cross terms. Example color terms might be: ($b_1 = 0.06, v_1 = -0.04, r_1 = -0.08, i_1 = 0.01$). You'll need to get them from Weidong.

so equation 9 actually stands for $N-1$ separate equations that will each give you a standard magnitude for the supernova at time t. So what you want to do, is pick the brightest standard stars and take the average of the values they give for $B_{sn}(t,i)$, and use that to calculate $B_{sn}(t)$ that does not depend on the particular standard stars used. For example, if you used all the standard stars, then $B_{sn}(t)$ is given by:

$$B_{sn}(t) = \left(\frac{1}{N-1}\right) \sum_{i=2}^{N} B_{sn}(t,i) \tag{10}$$

But if, for example, you only wanted to use the brightest secondary standard stars with indices 2, 3, 5, 6, and 8, then you would get $B_{sn}(t)$ from:

$$B_{sn}(t) = \left(\frac{B_{sn}(t,2) + B_{sn}(t,3) + B_{sn}(t,5) + B_{sn}(t,6) + B_{sn}(t,8)}{5}\right) \tag{11}$$

In general, the transformation equations for all bands that give you the standard supernova magnitude as a function of time t as calculated from secondary standard star i are given by: [55]

$$U_{sn}(t,i) = U_s(i) + u_{sn}(t) - u_s(t,i) + u_1\Big[\{u_{sn}(t) - v_{sn}(t)\} - \{u_s(t,i) - v_s(t,i)\}\Big] \tag{12}$$

$$B_{sn}(t,i) = B_s(i) + b_{sn}(t) - b_s(t,i) + b_1\Big[\{b_{sn}(t) - v_{sn}(t)\} - \{b_s(t,i) - v_s(t,i)\}\Big] \tag{13}$$

$$V_{sn}(t,i) = V_s(i) + v_{sn}(t) - v_s(t,i) + v_1\Big[\{b_{sn}(t) - v_{sn}(t)\} - \{b_s(t,i) - v_s(t,i)\}\Big] \tag{14}$$

$$R_{sn}(t,i) = R_s(i) + r_{sn}(t) - r_s(t,i) + r_1\Big[\{v_{sn}(t) - r_{sn}(t)\} - \{v_s(t,i) - r_s(t,i)\}\Big] \tag{15}$$

$$I_{sn}(t,j) = I_s(j) + i_{sn}(t) - i_s(t,j) + i_1\Big[\{v_{sn}(t) - i_{sn}(t)\} - \{v_s(t,j) - i_s(t,j)\}\Big] \tag{16}$$

Using these equations and averaging for the appropriate standard stars will give you the data to produce your standard supernova light curves. All that remains is to write programs to plot the data (magnitude vs. Julian date) and put the photometric data (plus errors) into a table. The plot of your standard light curve and the table of photometric data are your final results in publishable form. Congratulations! You're done!

---

[55] Note that in the final I band equation the standard star index j is used rather than i to avoid confusion with the fact that it is for the I 9or i) band.

8. Conclusion

If you've made it through this primer, you're about as crazy as I was for writing it. I started out with the goal of writing the document that I would have liked to have been handed when I first started trying to do photometry. Personally, I did the supernova search for a year starting in Spring 1999, then I went to Australia in Spring 2000 and when I came back in the Fall of 2000, I ran into Alex in the copy room and he asked me if I wanted to try doing photometry. At that point, I wasn't even sure that I wanted to continue with Astronomy, but when he asked me to re-join the group and start a new project, it sounded like a pretty good idea at the time. At that point, I had little experience using Unix and only a tiny amount of IRAF from the supernova search. Even basic things like how to log in remotely or make a new directory, let alone make a plot of my final data, were completely foreign to me. I got some help here and there from Maryam Modjaz and Weidong, but I was so far behind that as a result, I was frustrated and I languished during the semester and hardly did any work for the photometry for over a year, I believe.

I didn't really do anything of note until the Summer of 2001, when Amir Aazami and I were both around and we worked together to do photometry on a practice supernova (sn2000cx). It was tremendously motivating to work with another student and I'm convinced that this is the only way to really learn these things. I especially feel that way after having done the Infrared Astronomy Lab, where I learned a ridiculous amount, and a lot that was directly relevant to this research, mainly by working with other students and bashing our heads against these things. But after working with Amir that summer, I had a busy Fall 2001 semester (mainly due to the 40 hour a week Astro Lab...which I still highly recommend for the sheer intellectual insanity of it), didn't do any photometry, and forgot everything. It wasn't even until late this semester, in Spring 2002, when I had already graduated and been accepted to graduate school that I even had time to finish a non-practice supernova on my own (sn1998dh).

It's really hard to get motivated to look back at your notes and re-learn everything, and especially when there weren't any other students to work with at first. But when I started to work a bit with Brandon Swift and Marina Papenkova, teaching them what I had learned, that was plenty of motivation. Personally, I think the best way to get motivated to learn any esoteric task is to work with other people, because otherwise, you'll get incredibly frustrated and kill yourself over whether the stupid command is ":/..%" or ":/%..". And chances are that if there are a few people around, somebody will remember the command. That's probably the best advice I can give you. When you want to learn photometry, or IDl, or any programming at all, sit down with a few students from the group, log in remotely from 705, and do your thing.

In the end, my hope is that this document will serve as a reference that covers a huge amount of what you need to do, and act as a good supplement to the knowledge of Weidong and Alex and the other students in the group. Personally, if I had started doing photometry earlier, and had been given a primer like this, I believe I could have gotten comfortable with everything much sooner and ultimately done a lot more work. Thus, in that spirit, I hope to save all of you a lot of the trouble and frustration that I ran into, and maybe help allow you to keep your sanity. But again, the best way is to work with other people. Social collaboration in science should be stressed much more than it is. In the end, if you have a question, its much easier just to ask somebody who know rather than page through a gigantic manual. If nobody knows, check the manual. If you're just starting out, refer to the manual. If the manual ever fails, ask Weidong. He knows everything. That's my story and I'm sticking to it. Thank you, and good night.

Sincerely,
Andrew Samuel Friedman
June 11, 2002

P.S. By the way, it took forever every time I had to compile this thing in LaTeX. It was seriously ridiculous. In case you're wondering, I would guess that to write this approximately 100 page thing, all told, writing the text, getting the postscript files, formatting it in LaTeX, proofreading, etc... it took somewhere between 150 and 200 hours. It was worth it for me, because I'm crazy, and I hope it will be worth it for you to have for exactly the opposite reason.

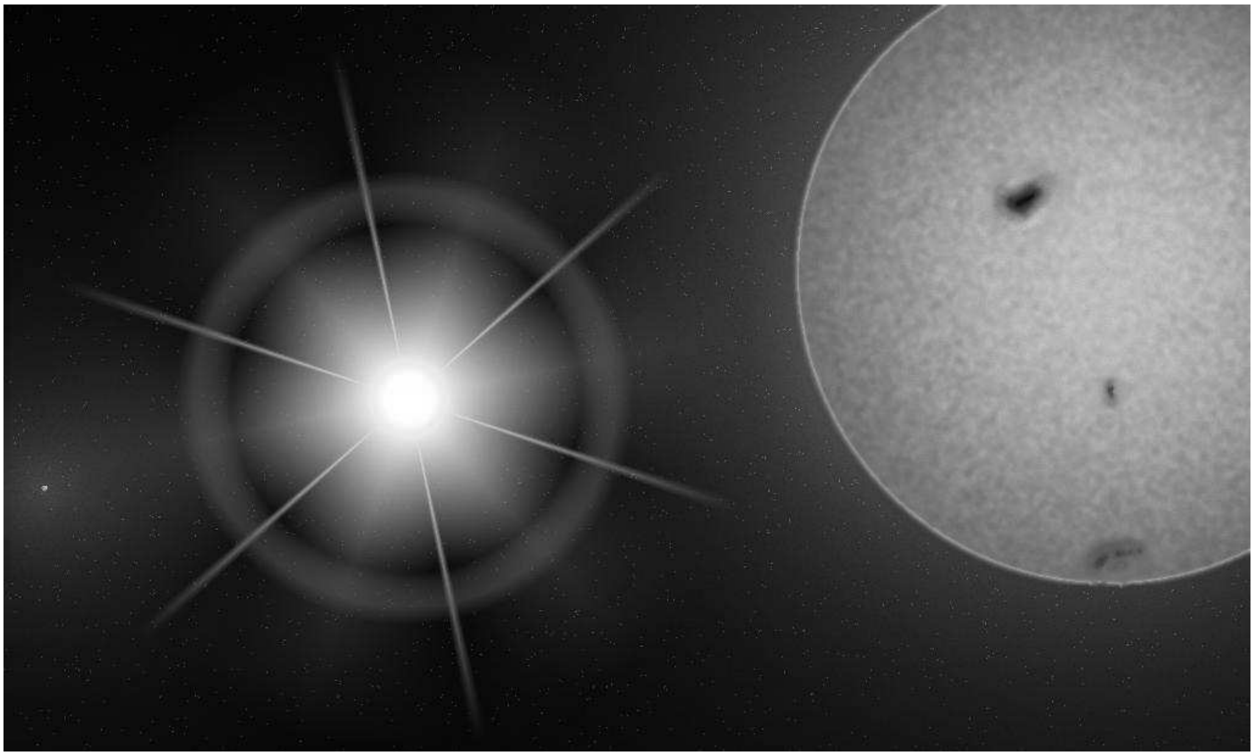

Fig. 16.— A supernova is what is happening to my head, as we speak. Artist Credit: Dirk Terrell.

9. APPENDIX

A. Renaming a Lot of Files in a Directory: vi and command line tricks.

If you want to systematically rename a bunch of files in a directory, you could do it manually, but if there are a large number of files, this can become quite time consuming. Thus, I have found it quite useful to take advantage of some cool vi and command line tricks. When you first organize your files, the systematic renaming process is taken care of by Weidong's program classfile, but you may find it necessary to rename lots of files together at some later point in the photometry process. For example, let's just say that you want to rename all the files in a directory with a .fts extension to a .fit extension. Here's how to do it in IRAF.

cl> ls *.fits > a	# Output all filenames that end in .fts to a file named a.
cl> cp a b	# Copy the file a to a new file named b.
cl> vi b	# Edit b in vi. First make sure it only includes files to be renamed
:1,$ s/fts/fit/g	# Search for all occurrences of "fts" and replace them with "fit"
cl> !paste a b > c	# Paste files a and b together as adjacent columns to a new file c
cl> vi c	# Edit c in vi
:1,$ s/∧/*mv* /	# Search for the beginning of each line and replace it with "mv "
	# The file c should now look like a list of renaming commands of
	# the form **mv** *filename*.**fts** *filename*.**fit**
cl> cl < c	# Executes the list of commands in file c on the command line
	# which renames all .fts files to .fit
cl> ls	# Files should appear with new names
cl> rm a b c	# Once you're done, you can delete files a, b, and c to clean up if desired

Note that this process is completely general for whatever search and replace commands you use in vi (or emacs as well) to rename the files as desired. If you find that you have a need to systematically rename files to some more convenient and consistent form, this process is a neat trick that can save you a lot of time.

B. References to IDL and LaTeX tutorials

I did go over some of the basics of using the UNIX operating system and various IRAF software packages. I had also hoped to get into some details about how to program in IDL (Interactive Data Language) and write scientific papers using LaTeX, but it ended up being beyond the scope of this primer. As such, I can refer you to a few particularly useful Unix, IDL, and LaTeX tutorials that can be found on the ugastro website at:

http://ugastro.berkeley.edu/tutorials/tutorials.html

In particular, the LaTeX tutorial at the bottom of the webpage, *The Not So Short Introduction to LaTeX 2$_\varepsilon$* by Tobias Oetiker et. al is quite good.

C. Uploading and Downloading files with ftp

You might find it useful to transfer files back and forth between hercules and your ugastro account. In this setting, you can use ftp (File Transfer Protocol), ssh (Secure Shell), or scp (Secure Copy), which uses ssh. I won't go into detail about the second two, but I will show you how to use ftp. Assume I'm logged into hercules, locally or remotely and that I want to transfer some files from my hercules directory /iraf_andy to my ugastro directory /friedman/Photometry/POSTSCRIPT. Here is a typical ftp command line session with comments. I exclude a few unnecessary command line messages that the machine spits out.

hercules:~/iraf_andy> ftp ugastro.berkely.edu	# oops, I misspelled the server name
ftp: ugastro.berkely.edu: Unknown host	#
ftp> exit	# Let's try this again
hercules:~/iraf_andy> ftp ugastro.berkeley.edu	#
Connected to ugastro.berkeley.edu.	# All right!
Name (ugastro.berkeley.edu:weidong): friedman	# Enter your ugastro username
331 Password required for friedman.	#
Password:	# Enter your ugastro password
230 User friedman logged in.	#
Remote system type is UNIX.	#
Using binary mode to transfer files.	#
ftp> cd Photometry/POSTSCRIPT	# transfer the files to this directory
250 CWD command successful.	#
ftp> put B0812gal.gif	# put *filename* : transfer a single file
local: B0812gal.gif remote: B0812gal.gif	#
150 Binary data connection for B0812gal.gif (128.32.92.171,2907).	#
226 Transfer complete.	#
ftp> mput s1_B0812.gif s2_B0812.gif	# mput *filename1 filename2*
	# use mput to transfer multiple files
mput s1_B0812.gif? y	# Do you really want to transfer this?
150 Binary data connection for s1_B0812.gif (128.32.92.171,2908).	#
226 Transfer complete.	#
mput s2_B0812.gif? y	# Enter 'y' for yes and 'n' for no.
150 Binary data connection for s2_B0812.gif (128.32.92.171,2909).	#
226 Transfer complete.	#
ftp> exit	# All right, we're done. Enjoy ftp.
221 Goodbye.	#
hercules:~/iraf_andy>	#

Table 23: A typical ftp command line session with comments. ftp is a useful way to transfer files between remote computers. The reverse process could easily be started when logged into ugastro by typing "ftp hercules.berkeley.edu", after which you would need to enter Weidong's password.

D. Acknowledgments/References

I would first of all like to thank Alex Filippenko, for giving me the opportunity to work in a setting where I can't help but feel like I ended up in the most relaxed and supportive research group on campus. Even though I haven't sampled them all, who can compete with pizza lunch research meetings filled with unbounded discussions into theoretical astrophysics. Way back in 1999, Alex sat me down and ran through the whole cosmic antigravity thing, told me he had a spot to check images and the rest is history.

I would also like to thank Weidong Li for being a great research advisor and really taking the time to help me through the image checking, and especially the more complicated steps of the photometry process. And I would like to thank both Weidong and Alex for forgiving my extreme business in a few of these past semesters. And I can't forget the fact that I studied abroad in Australia for the semester in Spring 2000 and was able to rejoin the group upon my return, no questions asked.

I would also like to thank Maryam Modjaz (now at Harvard) for helping me out with all sorts of Photometry questions, and Amir Aazami (now at Cambridge) for working with me through the entire process of generating the light curves for 2000cx and most of 1998dh. I would also like to thank Brandon Swift for working with me as we wrote a bunch of helpful IDL code relevant to photometry. Both Brandon and Marina Papenkova were excellent students who I learned a great deal from in my efforts to pass on my knowledge of photometry before heading off to graduate school. Together, we worked on sn2000cx, sn1998dh, and sn2002ap, while I started photometry on sn1998ef, but may leave it for other students to finish.

I would also like to thank James Graham and Nate McCrady, professor and GSI for the Astronomy 122 lab, in which I learned a tremendous amount that had a direct impact on my research. But who I really learned the most from were my fellow lab partners: Lee Huss, Christina Lee, Jim Brenna, Lindsay Pollock, Amy Jordan, Shane Bussman, Eric Nielsen, Kirsten Howley, Lauren Lau, John Potapenko

I would also like to thank the many other group members for helpful discussions and interactions during group meetings, some still with the group and some having moved on. I'm sure I'm forgetting someone, but I would like to thank Ryan Chornock, Doug Leonard, Adam Riess, Tom Matheson, Ed Moran, Ben Oppenheimer, Tom Sheffler, Eve Halderson, Jennifer King, Brandon Swift, and Marina Papenkova. I would also like to thank Richard Muller, my physics H195AB advisor, for allowing me to complete the physics honors thesis while still doing research in astronomy.

And finally, a special thanks to Weidong Li, Lee Huss, Brandon Swift, and Marina Papenkova for looking over the document at various stages and offering helpful suggestions. Thanks to Weidong and Lee especially for proofreading the document both for content and typos.

And last but not least, to my friends and family for understanding (or at least trying to understand) why I would put myself through the hell of writing this $\sim$100 page monster.

SELECTED REFERENCES

1. Apogee Instruments, CCD University: http://www.ccd.com/ccdu.html

2. Barnes, Jeanette *A Beginner's Guide to Using IRAF: IRAF Version 2.10*, Central Computer Services, National Optical Astronomy Observatories (NOAO), Tuscon Arizona, August 1993.

3. DaCosta, G.S. *Basic Photometric Techniques.*, Astronomical CCD Observing and Reduction Techniques, ASP Conference Series, Ed. Steve B. Howell, Vol. 23, 1992.

4. Finkbeiner, Douglas. *CCD Basics*, UC Berkeley Astronomy Department, Fall 2000.

5. Gilliland, Ronald L. *Details of Noise Sources and Reduction Processes*, Astronomical CCD Observing and Reduction Techniques, ASP Conference Series, Ed. Steve B. Howell, Vol. 23, 1992.

6. Graham, James R. *Lab 1: Photon Counting with a Photomultiplier Tube: Statistics of Light.*, UC Berkeley Undergraduate Infrared Astronomy Laboratory, Fall 2001.

7. Graham, James R. *Lab 2: Part 1: What is the Faintest Detectable Light Source?*, UC Berkeley Undergraduate Infrared Astronomy Laboratory, 9/14/2001.

8. Graham, James R. *Lab 2: Part 2: The Charge Coupled Device.*, UC Berkeley Undergraduate Infrared Astronomy Laboratory, 9/14/2001.

9. Graham, James R. *Lab 3: Photometry With and IR Camera.*, UC Berkeley Undergraduate Infrared Astronomy Laboratory, Fall 2001.

10. Graham, James R. *Lab 4: Imaging Obscured Star Clusters in the Infrared.*, UC Berkeley Undergraduate Infrared Astronomy Laboratory, Fall 2001.

11. Heasley, J.N. *Point-Spread Function Fitting Photometry*, CCD Precision Photometry Workshop, ASP Conference Series, Eds. Eric R. Craine et. al , Vol. 189, 1999.

12. Heiles, Carl. *A Short Unix Primer*, UC Berkeley Astronomy Department, 1/13/2001.

13. Howell, Steve B. *The Handbook of CCD Astronomy*, Cambridge University Press, 2000.

14. Lamport, Leslie *LaTeX: A User's Guide and Reference Manual*, Addison Wesley, Mass., 1994.

15. Massey, Philip & Davis, Lindsey E. *A User's Guide to Stellar CCD Photometry with IRAF*, 4/15/1992

16. Massey, Philip & Jacoby, George H. *CCD Data: The Good, the Bad, and the Ugly.*, Astronomical CCD Observing and Reduction Techniques, ASP Conference Series, Ed. Steve B. Howell, Vol. 23, 1992.

17. McCrady, Nate. *Lab 5: Surface Brightness and Extragalactic Astronomy*, UC Berkeley Undergraduate Infrared Astronomy Laboratory, Fall 2001.

18. Modjaz, Maryam *Doing Photometry in IRAF for well-isolated Supernovae*, UC Berkeley, 9/15/2000

19. Oetiker, Tobias et. al *The Not So Short Introduction to LaTeX 2_ε: Or LaTeX 2_ε in 95 minutes*, Version 3.20, 8/9/2001.

20. Robbins, Arnold *UNIX in a Nutshell: A Desktop Quick Reference for SVR4 and Solaris 7*, O'Reilly, 1999.

21. Space Art Websites, Dana Berry: http://spaceflightnow.com/news/n0102/09dwarf/

22. Space Art Websites, Dan Durda: http://www.boulder.swri.edu/ durda/paintings.html

23. Space Art Websites, Dirk Terrell: http://www.boulder.swri.edu/~terrell/dtart.htm

24. Todino, Grace et. al. *Learning the Unix Operating System: 5th Edition: A Concise Guide for the New User*, O'Reilly 2001.

25. Unix, IDL , and LaTeXTutorials From ugastro: http://ugastro.berkeley.edu/tutorials/tutorials.html

26. Various Authors, *IDL Users Guide: Interactive Data Language Version 4*, Research Systems Inc. 1995.

SELECTED REFERENCE TEXTBOOKS

1. Carroll Bradley W. and Ostlie, Dale A. *"An Introduction to Modern Astrophysics"*, Addison Wesley, 1996.

2. Pasachoff, Jay, and Filippenko Alex, *The Cosmos: Astronomy in the New Millenium*, Harcourt College Publishers, 2001.

3. Phillips, A.C. *"The Physics of Stars"* 2nd. ed. John Wiley & Sons, New York, 1999.

4. Shu, Frank. *The Physical Universe.* University Science Books, Sausalito, CA, 1980.

5. Taylor, John R., *An Introduction to Error Analysis.* University Science Books, Sausalito, CA, 1997, 2nd ed.

6. Woan, G. *The Cambridge Handbook of Physics Formulas* Cambridge University Press, 2000.

SELECTED REFERENCE PAPERS/PUBLICATIONS

1. Li W.D., Filippenko A.V., Treffers R.R., Friedman A., et. al., *"The Lick Observatory Supernova Search"* Cosmic Explosions, 2000, ed. S. Holt and W.W. Zheng, New York, American Institute of Physics. pp. 103-105
2. Li W.D., Filippenko A.V, and Friedman, A. *''Photometric Observations of 30 Nearby Type Ia Supernovae."* Paper In progress.
3. Friedman, A. *"The Lick Observatory Supernova Search: Type Ia Supernovae, Cosmology, and the Accelerating Universe."*, UC Berkeley, Senior Honors Thesis in Physics, Research Advisors: Professor Alex Filippenko (Astronomy), and Professor Richard Muller (Physics), December 2001
4. In April 2001, as part of my honors thesis, I presented a poster at the Berkeley Undergraduate Research Poster presentation organized by the Society of Physics Students, discussing the accelerating universe results and my role in the supernova search.
5. Friedman, A. *"Lab 1: Photon Statistics".* UC Berkeley Undergraduate Infrared Astronomy Laboratory, Professor James Graham, 9/11/2001
6. Friedman, A. *"Lab 2: Detecting Faint Signals".* UC Berkeley Undergraduate Infrared Astronomy Laboratory, Professor James Graham, 9/25/2001
7. Friedman, A. *"Lab 3: Photometry With The Leuschner Infrared Camera".* UC Berkeley Undergraduate Infrared Astronomy Laboratory, Professor James Graham, 10/19/2001
8. Friedman, A. *"Lab 4: Infrared Observations of the NGC1333 Star Forming Region".* UC Berkeley Undergraduate Infrared Astronomy Laboratory, Professor James Graham, 11/13/2001
9. Friedman, A. *"Lab 5: Extragalactic Astronomy and the Intensity Profiles of Galaxies".* UC Berkeley Undergraduate Infrared Astronomy Laboratory, Professor James Graham, 12/7/2001

SELECTED REFERENCE PAPERS

1. Filippenko A.V., Riess A.G. *"Results from the High-Z Supernova Search Team."*, Proceedings of the 3rd International Symposium on Sources and Detection of Dark Matter in the Universe (DM98), Feb. 1998, ed. D. Cline and Phys.Rept. 307 (1998) 31-44,
2. Filippenko A.V., Riess A.G. *"Evidence From Type Ia Supernova for an Accelerating Universe."*, Second Tropical Workshop on Particle Physics and Cosmology: Neutrino and Flavor Physics, ed. J. F. Nieves (New York: American Instit. of Phys.) Aug 2000
3. Graham, James, *"An Infrared Camera for Leuschner Observatory and the Berkeley Undergraduate Astronomy Lab"*, PASP, 113:607-621, 2001 May

4. Li W.D., Filippenko A.V. et. al. *"The Unique Type Ia Supernova 2000cx in NGC 524."*, PASP, Vol. 113, Issue 788, pp. 1178-1204. October 2001

5. Li, W.D., Filippenko, A.V,. Riess, A.G., Hu T.Y., Qiu, Y.L. *"A High Peculiarity Rate for Type Ia SNe".* In Cosmic Explosions, 2000. eds. S.S. Holt W.W. Zhang, New York, American Institute of Physics

6. Livio, M. *"The Progenitors of Type Ia Supernovae."*, Type Ia Supernovae: Theory and Cosmology, Cambridge Univ. Press, 27 July 2000

7. Livio, M. *"Type Ia Supernovae and their implications for cosmology."*. Proceedings of The greatest Explosions since the Big Bang: Supernovae and Gamma-Ray Bursts, May 1999 at STSCI,. eds. M. Livio, N. Panagia and K. Sahu, Cambridge Univ. Press, 2001

8. Modjaz, M., Li W.D., Filippenko A. V. Treffers R. R. *"High-Quality Optical Light Curves of Supernovae with the Katzman Automatic Imaging Telescope.* B. A. A. S., 196, 0903, (2000)

9. Modjaz, M. et. al. *The Subluminous Type Ia Supernova 1998de in NGC 252*, PASP, Vol. 113, Issue 781, pp. 308-325. March 2001

10. Perlmutter S. et. al. *"Cosmology From Type Ia Supernovae."*, Bull.Am.Astron.Soc. 29 (1997) 1351

11. Riess, A.G. , Filippenko A.V. et. al. *"Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant."*, Astron.J. 116 (1998) 1009-1038

12. Riess, A.G. et. al. *"Test of the Accelerating Universe with Near Infrared Observations of a High Redshift Type Ia Supernovae."*, 2000, ApJ 536, 62

13. Riess, A. G. et al., 2001, *"The Farthest Known Supernova: Support for an Accelerating Universe and a Glimpse of the Epoch of Deceleration."*, ApJ, Vol. 560, Issue 1, pp. 49-71.

14. Schmidt, et. al. *"The High-Z Supernova Search: Measuring Cosmic Deceleration and Global Curvature of the Universe Using Type Ia Supernovae."*, Astrophys.J. 507 (1998) 46-63