Tools for discovering and characterizing extrasolar planets

PhD Dissertation

Written by: András Pál

Loránd Eötvös University, Faculty of Sciences
PhD School of Physics
PhD Program of Particle- and Astrophysics

Head of PhD School of Physics: Prof. Zalán Horváth
Head of the PhD program: Prof. Ferenc Csikor

Thesis advisor: Dr. Gáspár Bakos,
postdoctoral fellow of the National Science Foundation
Harvard-Smithsonian Center for Astrophysics
Supervisor: Prof. Bálint Érdi, professor
Department of Astronomy
Loránd Eötvös University

Loránd Eötvös University
Budapest, Hungary
2009

Foreword

Among the group of extrasolar planets, transiting planets provide a great opportunity to obtain direct measurements for the basic physical properties, such as mass and radius of these objects. These planets are therefore highly important in the understanding of the evolution and formation of planetary systems: from the observations of photometric transits, the interior structure of the planet and atmospheric properties can also be constrained. The most efficient way to search for transiting extrasolar planets is based on wide-field surveys by hunting for short and shallow periodic dips in light curves covering quite long time intervals. These surveys monitor fields with several degrees in diameter and tens or hundreds of thousands of objects simultaneously. In the practice of astronomical observations, surveys of large field-of-view are rather new and therefore require special methods for photometric data reduction that have not been used before. Since 2004, I participate in the HATNet project, one of the leading initiatives in the competitive search for transiting planets. Due to the lack of software solution which is capable to handle and properly reduce the yield of such a wide-field survey, I have started to develop a new package designed to perform the related data processing and analysis. After several years of improvement, the software package became sufficiently robust and played a key role in the discovery of several transiting planets. In addition, various new algorithms for data reduction had to be developed, implemented and tested which were relevant during the reduction and the interpretation of data.

In this PhD thesis, I summarize my efforts related to the development of a complete software solution for high precision photometric reduction of astronomical images. I also demonstrate the role of this newly developed package and the related algorithms in the case of particular discoveries of the HATNet project.

Contents

Chapter 1

Introduction

In the last two decades, the discovery and characterization of extrasolar planets became an exciting field of astronomy. The first companion that was thought to be an object roughly 10 times more massive than Earth, had been detected around the pulsar PSR1829-10 (Bailes, Lyne & Shemar, 1991). Although this detection turned out to be a false one (Lyne & Bailes, 1992), shortly after the method of detecting planetary companions involving the analysis of pulsar timing variations led to the successful confirmation of the multiple planetary system around PSR1257+12 (Wolszczan & Frail, 1992). The pioneering discovery of a planet orbiting a main sequence star was announced by Mayor & Queloz (1995). They reported the presence of a short-period planet orbiting the Sun-like star 51 Peg. This detection was based on precise radial velocity measurements with uncertainties at the level of meter per second. Both discovery methods mentioned above are based on the fact that all components in a single or multiple planetary system, including the host star itself, revolve around the common barycenter, that is the point in the system having inertial motion. Thus, companions with smaller masses offset the barycenter only slightly from the host star whose motion is detected, either by the analysis of pulsar timing variations or by radial velocity measurements. Therefore, such methods – which are otherwise fairly common among the investigation techniques of binary or multiple stellar systems – yielded success in the form of confirming planets only after the evolution of instrumentation. Due to the physical constraints found in these methods, the masses of the planets can only be constrained by a lower limit, while we even do not have any information on the sizes of these objects.

The discovery of 51 Peg b was followed by numerous other detections, mainly by the method of radial velocity analysis, yielding the discovery, for instance, of the first planetary system with two planets around 47 UMa (Butler & Marcy, 1996; Fischer et al., 2002), and the first multiple planetary system around υ And (Butler et al., 1999). Until the first photometric detection of planetary transits in the system of HD 209458(b) (Henry et al., 2000; Charbonneau et al., 2000), no radius estimations could be given to the detected

planets, and all of these had only lower limits for their masses. Transiting planets provide the opportunity to characterize the size of the planet, and by the known inclination of its orbit, one can derive the mass of the planet without any ambiguity by combining the results of transit photometry with the radial velocity measurements. The planetary nature of HD 209458b was first confirmed by the analysis of radial velocity variations alone. The first discovery based on photometric detection of periodic dips in light curves was the discovery of OGLE-TR-56b (Konacki et al., 2003). Since several scenarios can mimic events that have similar light curves to transiting planets, confirmation spectroscopy and subsequent analysis of radial velocity data is still necessary to verify the planetary nature of objects found by transit searches (Queloz et al., 2001; Torres et al., 2005).

Since the first identification of planetary objects transiting their parent stars, numerous additional systems have been discovered either by the detection of transits after a confirmation based on radial velocity measurements or by searching transit-like signals in photometric time series and confirming the planetary nature with follow-up spectroscopic and radial velocity data. The former method led to the discovery of transits for many well-studied systems, such as HD 189733 (planet transiting a nearby K dwarf; Bouchy et al., 2005), GJ 436 (Butler et al., 2004), HD 17156 (Fischer et al., 2007; Barbieri et al., 2007) or HD 80606 (the transiting planet with the longest known orbital period of ~ 111 days; Naef et al., 2001; Moutou et al., 2009). These planets with transits confirmed later on are found around brighter stars since surveys for radial velocity variations mainly focus on these. However, the vast majority of the currently known transiting extrasolar planets have been detected by systematic photometric surveys, fully or partially dedicated for planet searches. Such projects monitor either faint targets using telescopes with small field-of-view or bright targets involving large field-of-view optical instrumentation. Some of the projects focused on the monitoring of smaller fields are the Monitor project (Irwin et al., 2006; Aigrainet et al., 2007), Deep MMT Transit Survey (Hartman et al., 2008), a survey for planetary transits in the field of NGC 7789 by Bramich et al. (2005), the "Survey for Transiting Extrasolar Planets in Stellar Systems" by Burke et al. (2004), "Planets in Stellar Clusters Extensive Search" (Mochejska et al., 2002, 2006), "Single-field transit survey toward the Lupus" of Weldrake et al. (2008), the SWEEPS project (Sahu et al., 2006), and the Optical Gravitational Lensing Experiment (OGLE) (Udalski et al., 1993, 2002; Konacki et al., 2003). Projects monitoring wide fields are the Wide Angle Search for Planets (WASP, SuperWASP, see Street et al., 2003; Pollacco et al., 2004; Cameron et al., 2007), the XO project (McCullough et al., 2005, 2006), the Hungarian-made Automated Telescope project (HATNet, Bakos et al., 2002, 2004), the Transatlantic Exoplanet Survey (TrES, Alonso et al., 2004), the Kilodegree Extremely Little Telescope (KELT, Pepper, Gould & DePoy, 2004; Pepper et al., 2007), and the Berlin Extrasolar Transit Search project (BEST, Rauer et al., 2004). One should mention here the existing space-borne project, the *CoRoT* mission (Barge et al., 2008) and the *Kepler* mission,

launched successfully on 7 March 2009 (Borucki et al., 2007). Both missions are dedicated (in part time) to searching for transiting extrasolar planets. As of March 2009, the above mentioned projects announced 57 planets. 6 planets were found by radial velocity surveys where transits were confirmed after the detection of RV variations (GJ 436b, HD 149026b, HD 17156b, HD 80606b, HD 189733b and HD 209458b), while the other 51 were discovered and announced by one of the above mentioned surveys. The *CoRoT* mission announced 7 planets, for which 4 had published orbital and planetary data; the OGLE project reported data for 7 planets and an additional planet with existing photometry in the OGLE archive has also been confirmed by an independent group (Snellen et al., 2008); the Transatlantic Exoplanet Survey reported the discovery of 4 planets; the XO project has detected and confirmed 5 planets; the SWEEPS project found 2 planets; the SuperWASP project announced $14 + 1$ planets, however, 2 of them are known only from conference announcements; and the HATNet project has $10 + 1$ confirmed planets. The planet WASP-11/HAT-P-10b had a shared discovery, it was confirmed independently by the SuperWASP and HATNet groups (this common discovery has been denoted earliet by the $+1$ term). The HATNet project also confirmed independently the planetary nature of the object XO-5b (Pál et al., 2008c).

All of the above mentioned wide-field surveys involve optical designs that yield a field-of-view of several degrees, moreover, the KELT project monitors areas having a size of thousand square degrees (hence the name, "Kilodegree Extremely Little Telescope"). The calibration and data reduction for such surveys revealed various problems that were not present on the image processing of "classical" data (obtained by telescopes with fast focal ratios and therefore smaller field-of-view). Some of the difficulties that occur are the following. Even the calibration frames themselves have to be filtered carefully, in order to avoid any significant structures (such as patches of clouds in the flat field images). Images taken by fast focal ratio optics have significant vignetting, therefore the calibration process should track its side effects, such as the variations in the signal-to-noise level across the image. Moreover, fast focal ratio yields comatic aberration and therefore systematic spatial variations in the stellar profiles. Such variations make the source extraction and star detection algorithms not only more sensitive but also are one of the major sources of the correlated noise (or red noise) presented in the final light curves [1]. Due to the large field-of-view and the numerous individual objects presented in the image, the source identification and the derivation of the proper "plate solution" for these images is also a non-trivial issue. The photometry itself is hardened by the very narrow and therefore undersampled sources. Unless the effects of the undersampled profiles and the spatial motions of the stellar profiles are handled with care, photometric time series are affected by strong systematics. Due to the short fractional duration and the shallow flux decrease of the planetary transits, several thousands of in-

---

[1]The time variation of stellar profiles is what causes red noise

dividual frames with proper photometry are required for significant and reliable detection. Since hundreds of thousands of stars are monitored simultaneously during the observation of a single field, the image reduction process yields enormous amount of photometric data, i.e. billions of individual photometric measurements. In fact, hundreds of gigabytes up to terabytes of processed images and tabulated data can be associated in a single monitored field. Even the most common operations on such a large amount of data require special methods.

The Hungarian Automated Telescope (HAT) project was initiated by Bohdan Paczýski and Gáspár Bakos (Bakos et al., 2002). Its successor, the Hungarian-made Automated Telescope Network (Bakos et al., 2004) is a network of small class of telescopes with large field-of-view, dedicated to an all-sky variability survey and search for planetary transits. In the past years, the project has became one of the most successful projects in the discovery of almost one fifth of the known transiting extrasolar planets. After joining the project in 2004, the author's goal was to overcome the above mentioned issues and problems, related to the image processing of the HATNet data. In this thesis, the efforts for the development of a software package and its related applications in the HATNet project are summarized.

This PhD thesis has five chapters. Following the Introduction, the second chapter, "Algorithms and Software environment" discusses the newly developed and applied algorithms that form the basis of the photometry pipeline, and gives a description on the primary concepts of the related software package in which these algorithms are implemented. The third chapter, "HATNet discoveries" describes a particular example for the application of the software on the analysis of the HATNet data. This application and the discussion is related to the discovery of the planet HAT-P-7b, transiting a late F star on a quite tight orbit. The fourth chapter, "Follow-up observations" focuses on the post-discovery measurements (including photometric and radial velocity data) of the eccentric transiting exoplanetary system of HAT-P-2b. The goals, methods and theses are summarized in the fifth chapter.

Chapter 2

Algorithms and Software environment

In principle, *data reduction* or simply *reduction* is the process when the raw data obtained by the instrumentation are transformed into a more useful form. In fact, raw data can be analyzed during acquisition in order do modify the instrumentation parameters for the subsequent measurements[1]. However, in the practice of astronomical data analysis, all raw data are treated to be known in advance of the reduction process. Moreover, the term "more useful form" of data is highly specific and depends on our needs. Regarding to photometric exoplanetary studies, this "more useful form" means two things. First – as in the case of HATNet where the discoveries are based on long-term photometric time series –, reduction ends at the stage of analyzed light curves, where transit candidates are recovered by the result of this analysis. Second, additional high-precision photometry[2] yields precise information directly about the planet itself. One should mention here that other types of measurements involving advanced and/or space-borne techniques (for instance, near-infrared photometry of secondary eclipses) have same principles of the reduction. The basics of the reductions are roughly the same and such observations yield even more types of planetary characteristics, such as brightness contrast or surface temperature distribution.

The primary platform for data reduction is computers and the reduction processes are performed by dedicated software systems. As it was mentioned in the introduction, existing software solutions lack several relevant components that are needed for a consistent analysis of the HATNet data flow. One of our goals was to develop a software package that features

---

[1]For instance, in the case of HATNet, real-time astrometric guiding is used to tweak the mount coordinates in the cases when the telescope drifts away from the desired celestial position. This guiding basically uses the same algorithms and routines that are involved in the photometric reduction. Like so, simplified forms of photometry can be used in the case of follow-up measurements of exoplanetary candidate host stars: if light curve variations show unexpected signals, the observation schedule could be changed accordingly to save expensive telescope time.

[2]Combined with additional techniques, such as spectroscopy or stellar evolution modelling. The confirmation the planetary nature by radial velocity measurements is essiential.

all of the functionality required by the proper reduction of the HATNet and the related follow-up photometry. The package itself is named FI/FIHAT, referring to both the HATNet project as well as the invocation of the related individual programs.

In the first major chapter of this PhD thesis, I summarize both the algorithms and their implementations that form the base of the FI/FIHAT software package. Due to the difficulties of the undersampled and wide-field photometry, several new methods and algorithms should have been developed, tested and implemented that were missing from existing and available image reduction packages. These difficulties are summarized in the next section (Sec. 2.1) while the capabilities and related problems of existing software solutions are discussed in Sec. 2.2.

The following sections describe the details of the algorithms and methods, focusing primarily on those that do not have any known implementation in any publicly available and/or commercial software. Sec. 2.3 discusses the details of the calibration process, Sec. 2.4 describes how the point-like sources (stars) are detected, extracted and characterized from the images, the details of the astrometry and the related problems – such as automatic source identification and obtaining the plate solution – are explained in Sec. 2.5, the details of the image registration process is discussed in Sec. 2.6, Sec. 2.7 summarizes the problems related to the instrumental photometry, Sec. 2.8 describes the concepts of the "image subtraction" process, that is mainly the derivation of a proper convolution transformation between two registered images, Sec. 2.9 explains how can the photometry be optimally performed on convolved or subtracted images. Sec. 2.10 describes the major concepts of how the still remaining systematic light curve variations can be removed.

In Sec. 2.11, after the above listed description of the crucial steps of the whole image reduction and photometry process, I outline the major principles of the newly developed software package. This part is then followed by the detailed description of the individual components of the software package. And finally, the chapter ends with the practices about how this package can be used in order to perform the complete image reduction process.

2.1 Difficulties with undersampled, crowded and wide-field images

In this section we summarize effects that are prominent in the reduction of the HATNet frames when compared to the "classical" style of image reductions. The difficulties can be categorized into three major groups. These groups do represent almost completely different kind of problems, however, all of these are the result of the type of the survey. Namely, these problems are related to the *undersampled* property, the *crowding* of the sources that are the point of interest and the *large field-of-view* of the images. In this section we examine what
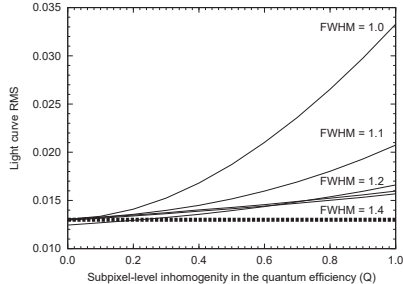
Figure 2.1: Plot showing the light curve scatter (rms, in magnitudes) of a mock star with various FWHMs and having a flux of $I = 10000$ (in electrons). The light curve rms is plotted as the function of the subpixel-level inhomogeneity, Q. Supposing a pixel structure where the pixel is divided to two rectangles of the same size, Q is defined as the difference in the normalized quantum efficiencies of these two parts (i.e. $Q = 0$ represents completely uniform sensitivity and at $Q = 1$ one of the parts is completely insensitive (that is typical for front-illuminated detector).

particular problems arise due to these properties.

2.1.1 Undersampled images

At a first glance, an image can be considered to be undersampled if the source profiles are "sharp". The most prevalent quantity that characterizes the sharpness of the (mostly stellar) profiles is the *full width at half magnitude* (FWHM). This parameter is the diameter of the contour that connects the points having the half of the source's peak intensity. Undersampled images therefore have (stellar) profiles with small FWHM, basically comparable to the pixel size. In the following, we list the most prominent effects of such a "small" FWHM and also check what is the practical limit below which this "small" is really small. In this short section we demonstrate the yields of various effects that are prominent in the photometry for stellar profiles with small FWHMs. All of these effects worsen the quality of the photometry unless special attention is made for their reduction.

Subpixel structure

The effect of the subpixel structure is relevant when the characteristic length of the flux variations becomes comparable to the scale length of the pixel-level sensitivity variations in the CCD detector. The latter is resulted mostly by the presence of the gate electrodes on the surface of the detector, that block the photons at certain regions of a given pixel. Therefore, this structure not only reduces the *quantum efficiency* of the chip but the signal depends on the centroid position of the incoming flux: the sharper the profile, the larger the dependence on the centroid positions. As regards to photometry, subpixel structure
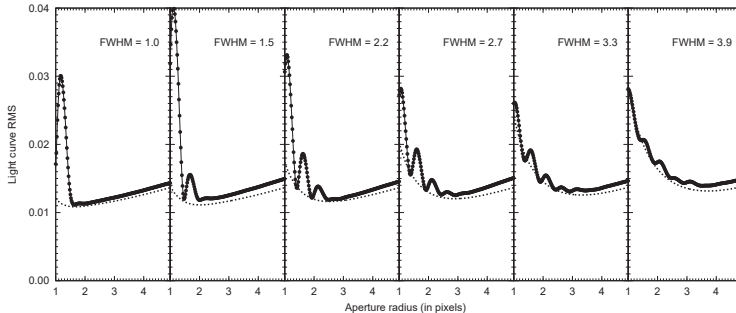
Figure 2.2: The graphs are showing the light curve scatters for mock stars (with 1% photon noise rms) when their flux is derived using aperture photometry. The subsequent panels shows the scatter for increasing stellar profile FWHM, assuming an aperture size between 1 and 5 pixels. The thick dots show the actual measured scatter while the dashed lines represent the lower limit of the light curve rms, derived from the photon noise and the background noise.

yields a non-negligible correlation between the raw and/or instrumental magnitudes and the fractional centroid positions. Advanced detectors such as *back-illuminated* CCD chips reduce the side effects of subpixel structure and also have larger quantum efficiency. Fig. 2.1 shows that the effect of the subpixel structure on the quality of the photometry highly dominates for sharp stars, where FWHM $\lesssim 1.2$ pixels.

Spatial quantization and the size of the aperture

On CCD images, aperture photometry is the simplest technique to derive fluxes of individual point sources. Moreover, advanced methods such as photometry based on PSF fitting or image subtraction also involve aperture photometry on the fit residuals and the difference images, thus the properties of this basic method should be well understood. In principle, aperture is a certain region around a source. For nearly symmetric sources, this aperture is generally a circular region with a pre-defined radius. Since the image itself is quantized (i.e. the fluxes are known only for each pixel) at the boundary of the aperture, the per pixel flux must be properly weighted by the area of the intersection between the aperture and the pixel. Aperture photometry is implemented in almost all of the astronomical data reduction software packages (see e.g. Stetson, 1987). As it is known from the literature (Howell, 1989), both small and large apertures yield small signal-to-noise ratio (SNR) or relatively high light curve scatter (or *root mean square*, rms). Small aperture contains small amount of flux therefore Poisson noise dominates. For large apertures, the background noise reduces the SNR ratio. Of course, the size of the *optimal aperture* depends on the total flux of the source as well as on the magnitude of the background noise. For fainter sources, this optimal
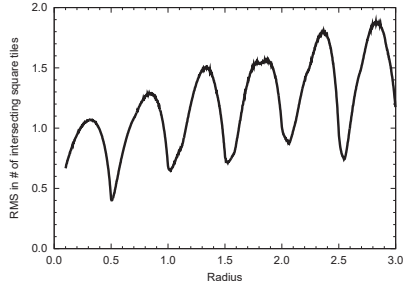
Figure 2.3: If circles with a fixed radius are drawn randomly and uniformly to a grid of squares, the number of intersecting squares has a well-defined scatter (since the number of squares intersecting the circle depends not only the radius of the circle but on the centroid position). The plot shows this scatter as the function of the radius.

aperture is smaller, approximately its radius is in the range of the profile FWHM, while for brighter stars it is few times larger than the FWHM (see also Howell, 1989). However, for very narrow/sharp sources, the above mentioned naive noise estimation becomes misleading. As it is seen in the subsequent panels of Fig. 2.2, the actual light curve scatter is a non-trivial oscillating function of the aperture size and this oscillation reduces and becomes negligible only for stellar profiles wider than FWHM $\gtrsim 4.0$ pixels. Moreover, a "bad" aperture can yield a light curve rms about 3 times higher than the expected for very narrow profiles. The oscillation has a characteristic period of roughly 0.5 pixels. It is worth to mention that this dependence of the light curve scatter on the aperture radius is a direct consequence of the topology of intersecting circles and squares. Let us consider a bunch of circles with the same radius, drawn randomly to a grid of squares. The actual number of the squares that intersect a given circle depends on the circle centroid position. Therefore, if the circles are drawn uniformly, this number of intersecting squares has a well defined scatter. In Fig. 2.3 this scatter is plotted as the function of the circle radius. As it can be seen, this scatter oscillates with a period of nearly 0.5 pixels. Albeit this problem is much more simpler than the problem of light curve scatter discussed above, the function that describes the dependence of the scatter in the number of intersecting squares on the circle radius has the same qualitative behavior (with the same period and positions of local minima). This is an indication of a non-trivial source of noise presented in the light curves if the data reduction is performed (at least partially) using the method of aperture photometry. In the case of HATNet, the typical FWHM is between $\sim 2 - 3$ pixels. Thus the selection of a proper aperture in the case of simple and image subtraction based photometry is essential. The methods intended to reduce the effects of this *quantization noise* are going to be discussed later on, see Sec. 2.10.
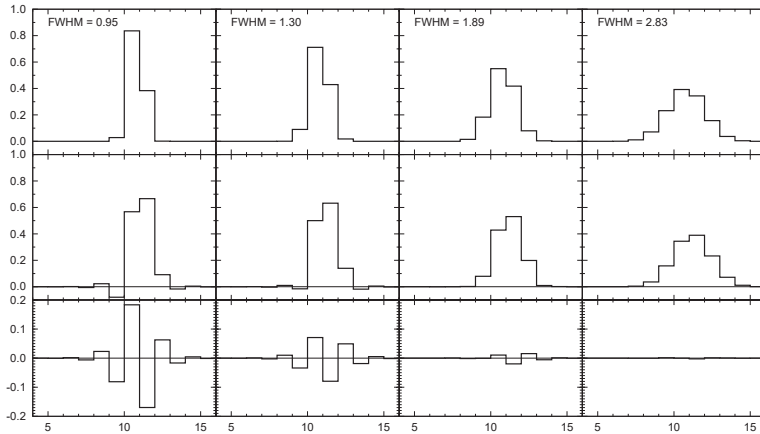
Figure 2.4: One-dimensional stellar profiles for various FWHMs, shifted using spline interpolation. The profiles on the upper stripe show the original profile while the plots in the middle stripe show the shifted ones. All of the profiles are Gaussian profiles (with the same total flux) and centered at $x_0 = 10.7$. The shift is done rightwards with an amplitude of $\Delta x = 0.4$. The plots in the lower stripe show the difference between the shifted profiles and a fiducial sampled profile centered at $x = x_0 + \Delta x = 10.7 + 0.4 = 11.1$.

Spline interpolation

As it is discussed later on, one of the relevant steps in the photometry based on image sub-traction is the *registration* process, when the images to be analyzed are spatially shifted to the same reference system. As it is known, the most efficient way to perform such a registra-tion is based on quadratic or cubic spline interpolations. Let us suppose a sharp structure (such as a narrow, undersampled stellar profile) that is shifted using a transformation aided by cubic spline interpolation. In Fig. 2.4 a series of one-dimensional sharp profiles are shown for various FWHMs between ~ 1 and ~ 3 pixels, before and after the transformation. As it can be seen well, for very narrow stars, the resulted structure has values smaller than the baseline of the original profile. For extremely sharp (FWHM ≈ 1) profiles, the magnitude of these undershoots can be as high as $10 - 15\%$ of the peak intensity. Moreover, the difference between the shifted structure and a fiducial profile centered on the shifted position also has a specific oscillating structure. The magnitude of such oscillations decreases dramatically if the FWHM is increased. For profiles with FWHM ≈ 3, the amplitude of such oscillation is about a few thousandths of the peak intensity (of the original profile). If the photometry is performed by the technique of image subtraction, such effects yield systematics in the photometry. Attempts to reduce these effects are discussed later on (see Sec. 2.9).
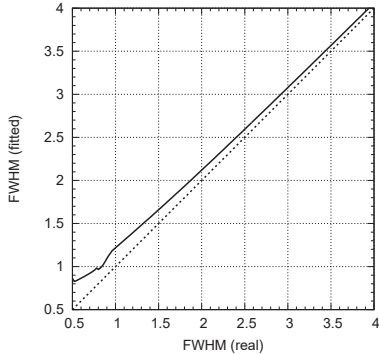
Figure 2.5: This plot shows how the profile FWHM is overestimated by the simplification of the fit. The continuous line shows the fitted FWHM if the model function is sampled at the pixel centers (instead of integrated properly on the pixels). The dashed line shows the identity function, for comparison purposes.

Profile modelling

Regarding to undersampled images, one should mention some relevant details of profile modelling. In most of the data reduction processes, stellar profiles detected on CCD images are characterized by simple analytic model functions. These functions (such as Gaussian or Moffat function) have a few parameters that are related to the centroid position, peak intensity and the profile shape parameters. During the extraction of stellar sources the parameters of such model functions are adjusted to have a best fit solution for the profile. In order to perform a self-consistent modelling, one should derive the integrated value of the model function to adjacent pixels and fit these integrals to the pixel values instead of sampling the model function on a square grid and fit these samples to the pixel values. Although the calculations of such integrals and its parametric derivatives[3] are computationally expensive, neglecting this effect yields systematic offsets in the centroid positions and a systematic overestimation of the profile size (FWHM). Since the plate solution is based on the individual profile centroid coordinates, such simplification in the profile modelling yields additional systematics in the final light curves[4] Moreover, precise profile modelling is essential in the reduction of the previously discussed spline interpolation side effect. As an example, in Fig. 2.5 we show how the fitted FWHM is overestimated by the ignorance of the proper profile modelling, if the profile model function is Gaussian.

---

[3]Parametric derivatives of the model functions are required by most of the fitting methods.

[4]For photometry, the final centroid positions are derived from the plate solution and a catalogue. Therefore, systematic variations in the plate solution indirectly yield systematic variations in the photometry and in the light curves.

2.1.2   Crowded images

Since the "CCD era", various dense stellar fields, such as globular clusters or open clusters are monitored for generic photometric analysis and variability search. The main problems of such crowded images are well known and several attempts have been done in order to reduce the side effects resulted from the merging profiles. In this section some of the problems are discussed briefly.

Merging and sensitivity to profile sharpness

Merging of the adjacent stellar profiles have basically two consequences in the point of photometry. First, it is hard to derive the background intensity and noise level around a given target star. Stars in the background area can be excluded by two ways: the pixels belonging to such profiles can be ignored either by treating them outliers or the proximity of other photometric centroids are removed from the set of background pixels. The second consequence of the profile merging is the fact that flux from adjacent stars is likely to be "flowed" underneath to the target aperture. Moreover, the magnitude of such additional flux depends extremely strongly on the profile FWHMs and therefore variations in the widths of the profiles cause significant increase in the light curve scatter.

Modelling

The modelling of stellar profiles, both by analytical model functions and empirical point-spread functions are definitely hardened in the case of merging sources. In this case the detected stars cannot be modelled separately, thus a joint fit should be performed simultaneously on all of the stars or at least on the ones that are relatively close to each other to have significant overlapping in the model functions. In the case of extremely crowded fields, sophisticated grouping and/or iterative procedures should be employed, otherwise the computation of the inverse matrices (associated with the parameter fitting) is not feasible (see also Stetson, 1987).

As we will see later on, the method of difference image photometry helps efficiently to reduce these side effects related to the crowdness of the images. However, it is true only for *differential* photometry, i.e. during the photometry of the reference frames[5] these problems still emerge.

---

[5]In principle, the method of differential photometry derives the flux of objects on a target image by adding the flux of the objects on a reference image to the flux of the residual on the image calculacted as the difference between the target and the reference image.

2.1.3 Large field-of-view

Additionally to the previously discussed issues, the large size of the field-of-view also introduces various difficulties.

Background variations

Images covering large field-of-view on the sky are supposed to have various background structures, such as thin cirrus clouds, or scattered light due to dusk, dawn or the proximity of the Moon or even interstellar clouds[6]. These background variations make impossible the derivation of a generic background level. Moreover the background level cannot be characterized by simple functions such as polynomials or splines since it has no any specific scale length. Because the lack of a well-defined background level, the source extraction algorithm is required to be purely topological (see also Sec. 2.4).

Vignetting, signal-to-noise level and effective gain

The large field-of-view can only be achieved by fast focal ratio optical designs. Such optical systems do not have negligible vignetting, i.e. the effective sensitivity of the whole system decreases at the corners of the image. In the case of HATNet optics, such vignetting can be as strong as 1 to 10. Namely, the total incoming flux at the corners of the image can be as small as the tenth of the flux at the center of the image. Although flat-field corrections eliminate this vignetting, the signal-to-noise ratio is unchanged. Since the latter is determined by the *electron count*, increasing the flux level reduces the effective *gain*[7] at the corner of the images. Since the expectations of the photometric quality (light curve scatter and/or signal-to-noise) highly depends on this specific gain value, the information about this yield of vignetting should be propagated through the whole photometric process.

Astrometry

Distortions due to the large field-of-view affects the astrometry and the source identification. Such distortions can efficiently be quantified with polynomial functions. After the sources are identified, the optimal polynomial degree (the order of the fit) can easily be obtained by calculating the unbiased fit residuals. For a sample series of HATNet images we computed these fit residuals, as it is shown in Table 2.1. It can easily be seen that the residuals do not

---

[6]Although interstellar clouds are steady background structures, in the point of the analysis of a *single* image, these cause the same kind of features on the image.

[7]The gain is defined as the joint electron/ADU conversion ratio of the amplifier and the A/D converter. A certain CCD camera may have a variable gain if the amplification level of the signal read from the detector can be varied before digitization.

Table 2.1: Typical astrometric residuals in the function of polynomial transformation order, for absolute and relative transformations. For absolute transformations the reference is an external catalog while for relative transformations, the reference is one of the frames.

Order	Absolute	Relative
1	$0.841 - 0.859$	$0.117 - 0.132$
2	$0.795 - 0.804$	$0.049 - 0.061$
3	$0.255 - 0.260$	$0.048 - 0.061$
4	$0.252 - 0.259$	$0.038 - 0.053$
5	$0.086 - 0.096$	$0.038 - 0.053$
6	$0.085 - 0.096$	$0.038 - 0.053$
7	$0.085 - 0.095$	$0.038 - 0.053$
8	$0.085 - 0.095$	$0.038 - 0.053$
9	$0.085 - 0.095$	$0.038 - 0.053$

decrease significantly after the $5 - 6$th order if an external catalogue is used as a reference, while the optimal polynomial degree is around $\sim 3 - 4$ if one of the images is used as a reference. The complex problem of the astrometry is discussed in Sec. 2.5 in more detail.

Variations in the profile shape parameters

Fast focal ratio optical instruments have significant comatic aberrations. The comatic aberration yields not only elongated stellar profiles but the elongation parameters (as well as the FWHMs themselves) vary across the image. As it was demonstrated, many steps of a complete photometric reduction depends on the profile sizes and shapes, the proper derivation of the shape variations is also a relevant issue.

Summary

In this section we have summarized various influences of image undersampling, crowdness and large field-of-view that directly or indirectly affects the quality of the photometry. Although each of the distinct effects can be well quantified, in practice all of these occur simultaneously. The lack of a complete and consistent software solution that would be capable to overcome these and further related problems lead us to start the development of a program designed for these specific problems.

In the next section we review the most wide-spread software solutions in the field of astronomical photometric data reduction.

Table 2.2: Comparison of some of the existing software solutions for astronomical image processing and data reduction. All of these software systems are available for the general public, however it does not mean automatically that the particular software is free or open source. This table focuses on the most wide-spread softwares, and we omit the "wrappers", that otherwise allows the access of such programs from different environments (for instance, processing of astronomical images in IDL use IRAF as a back-end).

Pros	Cons
IRAF[1]	• Not an open source software. Although the higher level modules and tasks are implemented in the own programming language of IRAF, the back-end programs have non-published source code. Therefore, many of the tasks and jobs are done by a kind of "black box", with no real assumption about its actual implementation.
• Image Reduction and Analysis Facility. The most commonly recognized software for astronomical data reduction, with large literature and numerous references.	
• IRAF supports the functionality of the package DAOPHOT[2], one of the most frequently used software solution for aperture photometry and PSF photometry with various fine-tune parameters.	• Old-style user interface. The primary user interface of IRAF follows the archaic designs and concepts from the eighties. Moreover, many options and parameters reflect the hardware conditions at that time (for instance, reading and writing data from/to tapes, assuming very small memory size in which the images do not fit and so on).
• IRAF is a complete solution for image analysis, no additional software is required if the general functionality and built-in algorithms of IRAF (up to instrumental photometry) are sufficient for our demands.	• Lack of functionality required by the proper processing of wide-field images. For instance, there is no particular effective implementation for astrometry or for light curve processing (such as transposing photometric data to light curves and doing some sort of manipulation on the light curves, such as de-trending).
ISIS[3].	• Not a complete software solution in a wider context. Additional software is required for image calibration, source detection and identification and also for the manipulation of the photometric results.
• Image subtraction package. The first software solution employing image subtraction based photometry.	
• The program performs all of the necessary steps related to the image subtraction algorithm itself and the photometry as well.	• Although this piece of software has open source codebase, the algorithmic details and some tricks related to the photometry on subtracted images are not documented (i.e. neither in the reference scientific articles nor in the program itself).
• Fully open source software, comes with some shell scripts (written in C shell), that demonstrate the usage of the program, as well as these scripts intend to perform the whole process (including image registration, a fit for convolution kernel and photometry).	• The kernel basis used by ISIS is fixed. The built-in basis involves a set of functions that can easily and successfully be applied on images with wider stellar profiles, but not efficient on images with narrow and/or undersampled profiles.
	• Some intermediate data are stored in blobs. Such blobs may contain useful information for further processing (such as the kernel solution itself), but the access to these blobs is highly inconvenient.
SExtractor[4].	• The primary goal of SExtractor was to be a package that focuses on source classification. Therefore, this package is not a complete solution for the general problem, it can be used only for certain steps of the whole data reduction.
• Source-Extractor. Widely used software package for extracting and classifying various kind of sources from astronomical images.	
• Open source software.	• Photometry is also designed for extended sources.
• Ability to perform photometry on the detected sources.	

---

[1] IRAF is distributed by the National Optical Astronomy Observatories, which are operated by the Association of Universities for Research in Astronomy, Inc., under cooperative agreement with the National Science Foundation. See also `http://iraf.net/`.

[2] DAOPHOT is a standalone photometry package, written by Peter Stetson at the Dominion Astrophysical Observatory (Stetson, 1987).

[3] ISIS is available from `http://www2.iap.fr/users/alard/package.html` with additional tutorials and documentation (Alard & Lupton, 1998; Alard, 2000).

[4] SExtractor is available from `http://sextractor.sourceforge.net/`, see also Bertin & Arnouts (1996).

2.2 Problems with available software solutions

In the past decades, several software packages became available for the general public, intended to perform astronomical data reductions. The most widely recognized package is the Image Reduction and Analysis Facility (IRAF), distributed by the National Optical Astronomy Observatories (NOAO). With the exception of photometry methods of image subtraction, many algorithms related to photometric data reductions have been implemented in the framework of IRAF (for instance, DAOPHOT, see e.g. Stetson, 1987, 1989). The first public implementation of the image convolution and the related photometry was given by Alard & Lupton (1998), in the form of the ISIS package. This package is focusing on certain steps of the procedure but it is not a complete solution for data reduction (i.e. ISIS alone is not sufficient if one should derive light curves from raw CCD frames, in this case other packages must be involved due to the lack of several functionalities in the ISIS package). The SExtractor package of Bertin & Arnouts (1996) intends to search, classify and characterize sources of various kind of shape and brightness. This program was designed for extragalactic surveys, however, it has several built-in methods for photometry as well. Of course, there are several other independent packages or wrappers for the previously mentioned ones[8]. Table 2.2 gives a general overview of the advantages and disadvantages of the previously discussed software packages. Currently, one can say that these packages alone *do not* provide sufficient functionality for the complete and consistent photometric reduction of the HATNet frames. In the following, we are focusing on those particular problems that arise during the photometric reductions of images similar to the HATNet frames and as of this writing, do not have any publicly available software solutions to overcome.

2.3 Calibration and masking

For astronomical images acquired by CCD detectors, the aim of the calibration process is twofold. The first goal is to reduce the effect of both the differential light sensitivity characteristics of the pixels and the large-scale variations yielded by the telescope optics. The second goal is to mark the pixels that must be excluded from the further data reduction since the previously mentioned corrections cannot be e performed because of various reasons. The most common sources of such reasons are the saturation and blooming of bright pixels, cosmic ray events or malfunctioning pixels (such as pixels with highly nonlinear response or with extraordinary dark current). Of course, some of these effects vary from image to image (e.g. saturation or cosmic ray events) while other ones (such as nonlinear pixels) have

---

[8]These wrappers allow the user to access functionalities from external data processing environments. For instance, the astronomical reduction package of the IDL environment uses the IRAF as a back-end, or the package PyRAF provides access to IRAF tasks within the Python language.

constant structure.

In this section the process of the calibration is described, briefly discussing the sensitivity corrections, followed by a bit more detailed explanation of the masking procedure (since it is a relevant improvement comparing to the existing software solutions). Finally, we show how these masks are realized and stored in practice. The actual software implementation related to the calibration process are described later in Sec. 2.12.

2.3.1 Steps of the calibration process

Basically, the calibration of all of the image frames, almost independently from the instrumentation, has been done involving bias, dark and flat images and overscan correction (where an appropriate overscan section is available on the detector). These calibration steps correct for the light sensitivity inhomogeneities with the exception of nonlinear responses, effects due to the dependence on the spatial and/or temporal variations in the telescope position or in the sky background[9] and second-order sensitivity effects[10]. In practice, the linear corrections provided by the classic calibration procedure are acceptable, as in the case of HATNet image calibrations.

Let us consider an image I and denote its calibrated form by $C(I)$. If the basic arithmetic operators between two images are defined as *per pixel* operations, $C(I)$ can be derived as

$$C(I) = \frac{I - O(I) - B_0 - (\tau[I]/\tau[D_0])D_0}{F_0/\|F_0\|}, \tag{2.1}$$

where $O(I)$ is the overscan level[11] of the image I; B_0, D_0 and F_0 are the *master* calibration images of bias, dark and flat, respectively. We denote the exposure time of the image x by $\tau[x]$. $\|x\|$ denotes the norm of the image x, that is simply the mean or median of the pixel values. In practice, when any of the above master calibration images does not exist in advance, one can substitute for these by zero, or in the case of flat images, by arbitrary positive constant value. The master calibration frames are the *per pixel* mean or median averages (with optional n-σ rejection) of individual frames:

$$C(B_i) = B_i - O(B_i), \tag{2.2}$$
$$B_0 = \langle C(B_i) \rangle, \tag{2.3}$$
$$C(D_i) = D_i - O(D_i) - B_0, \tag{2.4}$$

---

[9]Such as scattered light, multiple reflections in the optics or fringing yielded by the variations in the sky background spectrum

[10]such as the shutter effect

[11]Derived from the pixel values of the overscan area. The large scale structure of the overscan level is modelled by a simple function (such as spline or polynomial) and this function is then extrapolated to the image area.

$$D_0 = \langle C(D_i) \rangle, \tag{2.5}$$

$$C(F_i) = F_i - O(F_i) - B_0 - \frac{\tau[F_i]}{\tau[D_0]} D_0, \tag{2.6}$$

$$F_0 = \langle C(F_i) \rangle. \tag{2.7}$$

Equations (2.2), (2.4) and (2.6) clearly show that during the calibration of the individual bias, dark and flat frames, only overscan correction, overscan correction and a master bias frame, and overscan correction, a master bias and a master dark frame are used, respectively.

2.3.2 Masking

As it was mentioned earlier, pixels having some undesirable properties must be masked in order to exclude them from further processing. The FI/FIHAT package and therefore the pipeline of the whole reduction supports various kind of masks. These masks are transparently stored in the image headers (using special keywords) and preserved even if an independent software modifies the image. Technically, this mask is a bit-wise combination of Boolean flags, assigned to various properties of the pixels. In this paragraph we briefly summarize our masking method.

First, before any further processing and right after the readout of the images, a mask is added to mark the bad pixels of the image. Bad pixels are not only hot pixels but pixels where the readout is highly nonlinear or the readout noise is definitely larger than the average for the given detector. These *bad mask*s are determined after a couple of sky flats were acquired. Using sky flats for the estimation of nonlinearity and readout noise deviances are fairly good, since during dusk or dawn, images are exposed with different exposure times yielding approximately the same flux and all of the pixels have a locally uniform incoming flux. See Bakos (2004) for further details.

Second, all saturated pixels are marked with a *saturation mask*. In practice, there are two kind of effects related to the saturation: 1) when the pixel itself has an intensity that reaches the maximum expected ADU value or 2) if there is no support for anti-blooming in the detector, charges from saturated pixels can overflow into the adjacent ones during readout. These two types of saturation are distinguished in the *oversaturation mask* and *blooming mask*. If any of these mask are set, the pixel itself is treated as saturated. We note that this saturation masking procedure is also done before any calibration.

Third, after the calibration is done, additional masks can be added to mark the hot pixels (that were not corrected by subtracting the dark image), cosmic ray events and so on.

Actually, the latest version of the package supports the following masks:

- *Mask for faulty pixels.* These pixels show strong non-linearity. These masks are derived occasionally from the ratios of flat field images with low and high intensities.

- *Mask for hot pixels.* The mean dark current for these pixels is significantly higher than the dark current of normal pixels.

- *Mask for cosmic rays.* Cosmic rays cause sharp structures, these structures mostly resemble hot or bad pixels, but these does not have a fixed structure that is known in advance.

- *Mask for outer pixels.* After a geometric transformation (dilation, rotation, registration between two images), certain pixels near the edges of the frame have no corresponding pixels in the original frame. These pixels are masked as "outer" pixels.

- *Mask for oversaturated pixels.* These pixels have an ADU value that is above a certain limit defined near the maximum value of the A/D conversion (or below if the detector shows a general nonlinear response at higher signal levels).

- *Mask for blooming.* In the cases when the detector has no antiblooming feature or this feature is turned off, extremely saturated pixels causes "blooming" in certain directions (usually parallel to the readout direction). The A/D conversion value of the blooming pixels does not reach the maximum value of the A/D conversion, but these pixels also should be treated as somehow saturated. The "blooming" and "oversaturated" pixels are commonly referred as "saturated" pixels, i.e. the logical combination of these two respective masks indicates pixels that are related to the saturation and its side effects.

- *Mask for interpolated pixels.* Since the cosmic rays and hot pixels can be easily detected, in some cases it is worth to replace these pixels with an interpolated value derived from the neighboring pixels. However, these pixels should only be used with caution, therefore these are indicated by such a mask for the further processes.

We found that the above categories of 7 distinct masks are feasible for all kind of applications appearing in the data processing. The fact that there are 7 masks – all of which can be stored in a single bit for a given pixel – makes the implementation quite easy. All bits of the mask corresponding to a pixel fit in a byte and we still have an additional bit. It is rather convenient during the implementation of certain steps (e.g. the derivation of the blooming mask from the oversaturated mask), since there is a temporary storage space for a bit that can be used for arbitrary purpose.

2.3.3 Implementation

The basic *per pixel* arithmetic operations required by the calibration process are implemented in the program `fiarith` (see Sec. 2.12.2), while individual operations on associated masks can be performed using the `fiign` program (Sec. 2.12.6). Although the distinct steps of

```
MASKINFO= '1 -32 16,8 -16 0,1:-2 -32 1,1 -2,1 -16 1,0:2 -1,1:3,3 -32 3,2'
MASKINFO= '-16 -3,1:4 0,1:3,3 -32 3,0 -3,3 -16 1,0:2 0,1:-2 -32 1,0 -1,2'
```
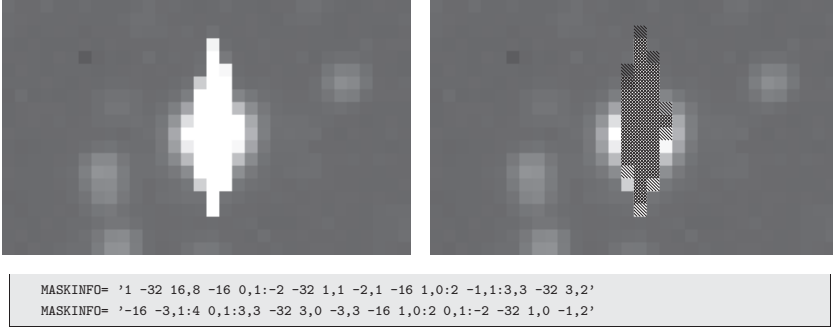
Figure 2.6: Stamp showing a typical saturated star. The images cover an approximately $8' \times 5'$ area (32×20 pixels) of the sky, taken by one of the HATNet telescopes. The blooming structure can be seen well. The left panel shows the original image itself. In the right panel, oversaturated pixels (where the actual ADU values reach the maximum of the A/D converter) are marked with right-diagonal stripes while pixels affected by blooming are marked with left-diagonal stripes. Note that most of the oversaturated pixels are also blooming ones, since their lower and/or upper neighboring pixels are also oversaturated. Such pixels are therefore marked with both left- and right-diagonal stripes. Since the readout direction in this particular detector was vertical, the saturation/blooming structure is also vertical. The ''MASKINFO'' blocks seen below the two stamps show how this particular masking information is stored in the FITS headers in a form of special keywords.

Value	Interpretation
T	Use type T encoding. $T = 0$ implies absolute cursor movements, $T = 1$ implies relative cursor movements. Other values of T are reserved for optional further improvements.
$-M$	Set the current bitmask to M. M must be between 1 and 127 and it is a bit-wise combination of the numbers 1, 2, 4, 8, 16, 32 and 64, for faulty, hot, cosmic, outer, oversaturated, blooming and interpolated pixels, respectively.
x, y	Move the cursor to the position (x, y) (in the case of $T = 0$) or shift the cursor position by (x, y) (in the case of $T = 1$) and mark the pixel with the mask value of M.
$x, y : h$	Move/shift the cursor to/by (x, y) and mark the horizontal line having the length of h and left endpoint at the actual position.
$x, y : -v$	Move/shift the cursor to/by (x, y) and mark the vertical line having the length of v and lower endpoint at the actual position.
$x, y : h, w$	Move/shift the cursor to/by (x, y) and mark the rectangle having a size of $h \times w$ and lower-left corner at the actual cursor position.

Figure 2.7: Interpretation of the tags found MASKINFO keywords in order to decode the respective mask. The values of M, h, v and w must be always positive.

the calibration can be performed by the appropriate subsequent invocation of the above two programs, a more efficient implementation is given by `ficalib` (Sec. 2.12.5), that allows fast evaluation of equation (2.1) on a large set of images. Moreover, `ficalib` also creates the appropriate masks upon request. The master calibration frames (referred as B_0, D_0

and F_0 in equation 2.1) are created by the combination of the individual calibration images (see equations 2.3, 2.5 and 2.7), involving the program `ficombine` (Sec. 2.12.4). See also Sec. 2.12.5 for more specific examples about the application of these programs.

As it was mentioned earlier, the masks are stored in the FITS header using special keywords. Since pixels needed to be masked represent a little fraction of the total CCD area, only information (i.e. mask type and coordinates) about these masked pixels are written to the header. By default, all other pixels are "good". A special form of run-length encoding is used to compress the mask itself, and the compressed mask is then represented by a series of integer numbers. This series of integers should be interpreted as follows. Depending on the values of these numbers, a virtual "cursor" is moved along the image. After each movement, the pixel under the cursor or a rectangle whose lower-left corner is at the current cursor position is masked accordingly. In Fig. 2.6 a certain example is shown demonstrating the masks in the case of a saturated star (from one of the HATNet images). The respective encoded masks (as stored literally in the FITS header) can be seen below the image stamps. The encoding scheme is summarized in Fig. 2.7. We found that this type of encoding (and the related implementation) provides an efficient way of storing such masks. Namely, the encoding and decoding requires negligible amount of computing time and the total information about the masking requires a few dozens from these "`MASKINFO`" keywords, i.e. the size of the FITS image files increases only by $3-5$ kbytes (i.e. by less than 1%).

2.4 Detection of stars

Calibration of the images is followed by detection of stars. A successful detection of star-like objects is not only important because of the reduction of the data but for the telescopes of HATNet it is used *in situ* for guiding and slewing corrections.

In the typical field-of-view of a HATNet telescope there are 10^4 – 10^5 stars with suitable signal-to-noise ratio (SNR), which are proper candidates for photometry. Additionally, there are several hundreds of thousands, or millions of stars which are also easy to detect and characterize but not used for further photometry. The HATNet telescopes acquire images that are highly crowded and undersampled, due to the fast focal ratio instrumentation ($f/1.8$ for the lenses used by the HAT telescopes). Because of the large field-of-view, the sky background does also vary rapidly on an ordinary image frame, due to the large-scale structure of the Milky Way, atmospheric clouds, differential extinction or another light scattering effects. Due to the fast focal ratio, the vignetting effects are also strong, yielding stars of the same magnitude to have different SNR in the center of the images and the corners. This focal ratio also results in stars with different shape parameters, i.e. systematically and heavily varying FWHM and elongation even for focused images (this effect is known as *comatic aberration*

or *coma*). These parameters may also vary due to the different sky conditions, e.g. airmass resulted also by the large FOV (12-15 degrees in the diameter).

Because of these, one should expect the following properties of a star detection and characterization algorithm that are thereafter able to overcome the above mentioned problems.

A) The method should be local both in the sense of pixel positions and in the intensity. Namely, the result of an object detection must not differ if one applies an affine transformation in the intensity or after a spatial shift of the image.

B) The method should not contain any characteristic scale length, due to the unpredictable scale length of the background. It also implies that there should not be any kind of "partitioning the image into blocks" during the detection, i.e. one should not expect that any of the above mentioned affects disappear if the image is divided into certain blocks and some quantities are treated as constants in such a block. Moreover, there is no "background" for the image, even one cannot do any kind of interpolation to determine a smooth background.

C) The algorithm should be fast. Namely, it is expected to be an $\mathcal{O}(N)$ algorithm, where $N = S_x \times S_y$, the total number of the image pixels. In other words, the computing time is expected to be nearly independent from the number and/or the density of the detected objects.

D) On highly crowded *and* undersampled images, stars should be distinguished even if they are very close to each other. Thus, the direct detection should not be preceded by a convolution with a kernel function, as it is done in the most common algorithms and software (e.g. as it is used in DAOPHOT/FIND, see Stetson, 1987). Although this preliminary convolution increases the detectability of low surface brightness object, in our case it would fuse nearby stars.

E) The algorithm may have as few as possible external fine-tune parameters.

F) The algorithm should explicitly assign the pixels to the appropriate detected objects.

G) Last but not least, the algorithm should work properly not only for undersampled and crowded images but for images acquired by "classic" types of telescopes where the average FWHM of the stars are higher and/or the number density is lower. Additionally, one may expect from such an algorithm to handle the cases of smeared or defocused images, even when the star profiles have "doughnut" shape, as well as the proper characterization of digitized photographic data (e.g. POSS/DSS).

In this section we give an algorithm that is suitable for the above criteria. Moreover, it is purely topological since considers only "less than" or "greater than" relations between ad-

jacent pixel intensities. Obviously, an algorithm that relies only the topology, automatically satisfies the conditions A and B above. The first part of this section discusses how can the image be partitioned to smaller partitions that are sets of cohesive pixels that belong to the same star (see also condition F above). The second part of the section describes how these partitions/stars can be characterized by a couple of numbers, such as centroid coordinates, flux, and shape parameters.

2.4.1 Image partitioning

Pixel links and equivalence classes

The first step of the detection algorithm is to define local pixel connections with the following properties. An ordinary pixel has 8 neighbors, and the number of neighbors is less only if the pixel is a boundary pixel (in this case there can be 5 or 3 neighbors) or if any of the neighboring pixels are excluded due to a mask of bad, hot or saturated pixel. Including the examined pixel with the coordinates of x and y, we select the one with the largest intensity from this set. Let us denote the coordinates of this pixel by $n_x(x, y)$ and $n_y(x, y)$. For a shorter notation, we introduce $\mathbf{x} = (x, y)$ and $\mathbf{n}(\mathbf{x}) = [n_x(x, y), n_y(x, y)]$. Obviously, $|n_x - x| \leq 1$ and $|n_y - y| \leq 1$, i.e. $\|\mathbf{n}(\mathbf{x}) - \mathbf{x}\|_\infty \leq 1$, where $\|\mathbf{x}\|_\infty = \max(|x|, |y|)$, the maximal norm. The derivation of this set of $\mathbf{n} = (n_x, n_y)$ points requires $\mathcal{O}(N)$ time. Second, we define $\mathbf{m}(\mathbf{x}) = [m_x(x, y), m_y(x, y)]$ for a given pixel by

$$\mathbf{m}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \mathbf{n}(\mathbf{x}) = \mathbf{x}, \\ \mathbf{m}(\mathbf{n}(\mathbf{x})) & \text{otherwise.} \end{cases} \qquad (2.8)$$

Note that this definition of $\mathbf{m}(\mathbf{x})$ is only a functional of the relation $\mathbf{x} \to \mathbf{n}(\mathbf{x})$: there is no need for the knowledge of the underlying neighboring and the partial ordering between pixels. This definition results a set of finite pixel links $\mathbf{x}, \mathbf{n}(\mathbf{x}), \mathbf{n}(\mathbf{n}(\mathbf{x})) \equiv \mathbf{n}^2(\mathbf{x}), \ldots$ where the length L of this link is the smallest value where $\mathbf{n}^L(\mathbf{x}) = \mathbf{n}^{L+1}(\mathbf{x}) = \mathbf{m}(\mathbf{x})$. Third, we define two pixels, say, $\mathbf{x}_1 = (x_1, y_1)$ and $\mathbf{x}_2 = (x_2, y_2)$ to be *equivalent* if $\mathbf{m}(\mathbf{x}_1) = \mathbf{m}(\mathbf{x}_2)$. This equivalence relation partitions the image into disjoint sets, equivalence classes. In other words, each equivalence contains links with the same endpoint. Let us denote these classes by C_i.

Each class is represented by the appropriate $\mathbf{m}_i \equiv \mathbf{m}(C_i)$ pixel, that is, by definition, a local maximum. Each equivalence class can be considered as a possible star, or a part of a star if the image was defocused or smeared. In Fig. 2.8, one can see stamps from a typical image obtained by one of the HATNet telescopes and the derived pixel links and the respective equivalence classes. In the figure, the mapping $\mathbf{x} \to \mathbf{n}(\mathbf{x})$ is represented by the $\mathbf{n}(\mathbf{x}) - \mathbf{x}$ vectors, originating from the pixel $\mathbf{x}$.
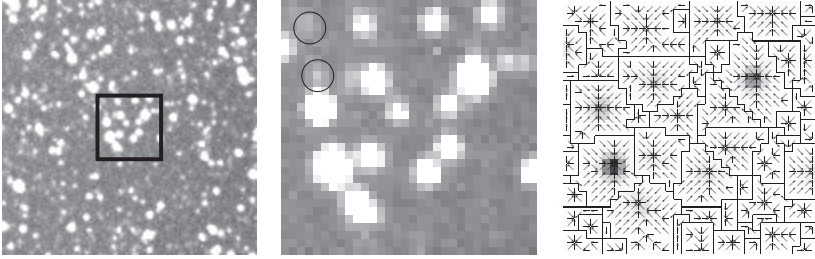
Figure 2.8: Left panel: a stamp of 128×128 pixels from a typical crowded HAT image, covering approximately $0.5° \times 0.5°$ area on the sky. Middle panel: the central are of the stamp shown in the left panel, covering approximately an area of $7' \times 7'$. This smaller stamp has a size of 32×32 pixels. Right panel: the links and equivalence classes generated from the smaller stamp. Note that even the faintest stars are detected and the belonging pixels form separated partitions (for an example see the stars encircled on the middle panel).

Background

Let us define the number of possible neighbors of a given pixel $\mathbf{x}$ by $K_0(\mathbf{x})$. As it was described above, in average it is 8, for boundary pixels it is 5 or 3, and it can be less if there are surrounding masked ones. The quantity $R(\mathbf{x})$ is defined by the cardinality of the set $\{\mathbf{x}' \in \text{Image} : \mathbf{n}(\mathbf{x}') = \mathbf{x}\}$. Let us also define $K(\mathbf{x}) = K_0(\mathbf{x}) + 1$ and $G(\mathbf{x})$ as the cardinality of

$$\{\mathbf{x}' \in \text{Image} : \|\mathbf{x}' - \mathbf{x}\|_\infty \leq 1 \text{ and } \mathbf{m}(\mathbf{x}') = \mathbf{m}(\mathbf{x})\}, \tag{2.9}$$

which is the number of surrounding pixels in the same class. For a given equivalence class C we can define its *background pixels* by

$$\mathbf{B}(C) = \{\mathbf{x}' \in C : R(\mathbf{x}') = 0 \text{ and } G(\mathbf{x}') < K(\mathbf{x}')\}. \tag{2.10}$$

This set of pixels are the boundary starting points of pixel links in this equivalence class. Note that this definition may not reflect the true background if there are merging stars in the vicinity. In such case these pixels are saddle points between two or more stars. However, the median of the pixel intensities in the set $\mathbf{B}(C)$ is a good assumption of the local background for the star candidate C, even for highly crowded images. For simplicity, let us denote the background of C by

$$b(C) \equiv \langle \mathbf{B}(C) \rangle. \tag{2.11}$$

Collection of subsets

The above definitions of equivalence classes and background pixels are quite robust ones, but still there are some demands for certain cases. First, in extremely crowded fields, the number of background pixels can be too small for a local background assumption. Second, the defocused or smeared stars may consist of several separate local maxima that yield distinct

equivalence classes instead of one cohesive set of pixels. To overcome these problems, we make some other definitions. An equivalence class C is *degenerated* if

$$R(\mathbf{m}(C)) < K(\mathbf{m}(C)). \tag{2.12}$$

In other words, degenerated partitions have local maxima on their boundary. For such a partition, one can define the two sets of pixels:

$$\mathbf{J}_1(C) = \{\mathbf{x}' \in \text{Image} \ : \ \|\mathbf{x}' - \mathbf{m}(C)\|_\infty = 1\}, \tag{2.13}$$

and

$$\mathbf{J}_2(C) = \{\mathbf{x}' \in \mathbf{J}_1(C) \ : \ \mathbf{m}(\mathbf{x}') \neq \mathbf{m}(C)\}. \tag{2.14}$$

Let us denote the location of the maximum of a given set $\mathbf{J}$ by

$$\mathbf{M}(\mathbf{J}) = \{\mathbf{x} \ : \ \forall \ \mathbf{x}' \in \mathbf{J} \ I(\mathbf{x}') \leq I(\mathbf{x}),\} \tag{2.15}$$

where $I(\mathbf{x})$ is the intensity of the pixel $\mathbf{x}$. Using the above definitions, we can coalesce this degenerated partition C with one or more other partitions by two ways. Obviously, $\mathbf{n}(\mathbf{m}(C)) = \mathbf{m}(C)$, so we re-define $\mathbf{n}(\mathbf{m}(C))$ by either

$$\mathbf{n}'_1(\mathbf{m}(C)) := \mathbf{M}(\mathbf{J}_1(C)) \tag{2.16}$$

if and only if $\mathbf{J}_1(C)$ is not the empty set *and* $\mathbf{m}[\mathbf{M}(\mathbf{J}_1(C))] \neq \mathbf{m}(C)$ or

$$\mathbf{n}'_2(\mathbf{m}(C)) := \mathbf{M}(\mathbf{J}_2(C)) \tag{2.17}$$

if and only if $\mathbf{J}_2(C)$ is not the empty set. Otherwise we do not affect $\mathbf{n}(\mathbf{m}(C))$. We note that the latter expansion may result in a larger amount of coalescing sets, i.e. in the former case it may happen that the maximum of the neighboring pixels fall into the same class while in the latter case we definitely excluded such cases (see the definition of $\mathbf{J}_2(C)$).

Prominence

In case of highly defocused star images, the PSF can be donut-shaped and a single star may have separated distinct (and not degenerated) maxima. To coalesce such equivalence classes, we define the discrete prominence, with almost the same properties as it is known from topography. The prominence of a mountain peak in topography (a.k.a. topographic prominence or autonomous height) is defined as follows. For every path connecting the peak to higher terrain, find the lowest point on that path, that is at a saddle point. The key saddle is defined as the highest of these saddles, along all connecting paths. Then the prominence is the difference between the elevation of the peak and the elevation of the key saddle. This definition cannot be directly applied to our discrete case, since the number of
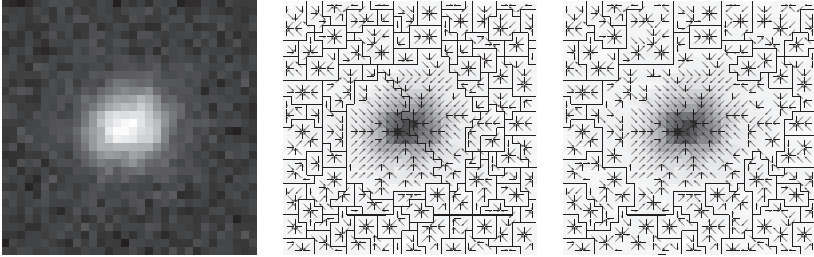
Figure 2.9: Left panel: a stamp of a star, covering 32×32 pixels from a typical blurred KeplerCam image. Middle panel: the links and equivalence classes generated from this stamp, using the basic algorithm without any coalescing. Right panel: the links and equivalence classes generated from the stamp, when the partitions with zero prominence are joined to their neighboring partitions.

possible connecting paths between two maxima is an exponential function of the number of the pixels, i.e. we cannot get an $\mathcal{O}(N)$ algorithm. Thus, we use the following definition for the key saddle $\mathbf{s}$ of an equivalence class C:

$$\mathbf{s}(C) \quad = \quad \{\mathbf{x} \in C \ : \ G(\mathbf{x}) < K(\mathbf{x}) \text{ and} \qquad\qquad (2.18)$$
$$\forall \, \mathbf{x}' \in C \ G(\mathbf{x}') < K(\mathbf{x}') \Rightarrow I(\mathbf{x}') \leq I(\mathbf{x})\}.$$

Thus, the *prominence* of this class is going to be

$$p(C) = I(\mathbf{m}(C)) - I(\mathbf{s}(C)). \qquad\qquad (2.19)$$

Note that $p(C)$ is always non-negative and if C is degenerated, $p(C)$ is zero. The *related class*es $\mathcal{R}(C)$ of C are defined as

$$\mathcal{R}(C) = \{C' \in \text{Classes} \ : \ \exists \mathbf{x}' \in C' \, \|\mathbf{x}' - \mathbf{s}(C)\| = 1\} \qquad\qquad (2.20)$$

We define the set of *parent classes* of C as the set

$$P(C) \quad = \quad \{C' \in \mathcal{R}(C) \ : \ \forall \, C'' \in \mathcal{R}(C) \qquad\qquad (2.21)$$
$$\mathbf{m}(C'') \leq \mathbf{m}(C') \text{ and } \mathbf{m}(C) < \mathbf{m}(C'')\}$$

The set of parent classes $P(C)$ can be empty if the class C is the most prominent one. If at least one parent class exists, the *relative prominence* of C is defined as

$$r(C) = \frac{p(C)}{I[\mathbf{m}(P(C))] - \langle \mathbf{B}(P(C)) \rangle}, \qquad\qquad (2.22)$$

and, by definition, it is always between 0 and 1. Since the classes with low relative prominences are most likely parts of a larger object that is dominated by the parent class (or, moreover, by the parent of the parent class and so on), we connect these low-prominence

classes to their parents below a critical relative prominence r_0. Namely, we alter $\mathbf{n}(\mathbf{s}(C))$ to one point of $P(C)$, say, $\mathbf{x}' \in P(C)$ where $\|\mathbf{s}(C) - \mathbf{x}'\| = 1$. Note that this algorithm for $r_0 = 0$ yields the same collection of partitions as the usage of the definition $\mathbf{J}_2(C)$ in the end of the previous subsection only for degenerated partitions.

2.4.2 Coordinates, shape parameters and analytic models

In the previous sections we have discussed how astronomical images can be partitioned in order to extract sets of pixels that belong to the same source. Now we describe how these partitions can be characterized, i.e. how can one determine the centroid coordinates, total flux of the source and quantify somehow the shape of the source.

Weighted mean and standard deviance

The easiest and fastest way to get some estimation on the centroid coordinates and the shape parameters of the source is to calculate the statistical mean and standard deviation of the pixel coordinates, weighted by the individual fluxes after background subtraction. Let us consider a set of pixels, $C = \{\mathbf{x}_i\}$, each of them has the flux (ADU value) of f_i, while the background level B of this source is calculated by using equation (2.11). Then the weighted coordinates are

$$\langle \mathbf{x} \rangle = \frac{\sum_i (f_i - B)\mathbf{x}_i}{\sum_i (f_i - B)}, \tag{2.23}$$

while the statistical standard deviation in the coordinates is the covariance matrix, defined as

$$\mathbf{S} = \frac{\sum_i (f_i - B)(\mathbf{x}_i - \langle \mathbf{x} \rangle) \circ (\mathbf{x}_i - \langle \mathbf{x} \rangle)}{\sum_i (f_i - B)}. \tag{2.24}$$

Let us denote the components of the matrix $\mathbf{S}$ by

$$\mathbf{S} = \begin{pmatrix} \Sigma + \Delta & \mathrm{K} \\ \mathrm{K} & \Sigma - \Delta \end{pmatrix}. \tag{2.25}$$

For objects that are not elongated, $\Delta = \mathrm{K} = 0$. It can be shown that for elongated objects, the semimajor axis of the best fit ellipse (to the contours) has a position angle of $\varphi = \frac{1}{2}\arg(\Delta, \mathrm{K})$ and an ellipticity of $\sqrt{\Delta^2 + \mathrm{K}^2}/\Sigma$. The size of the star profiles are commonly characterized by the "full width at half magnitude" (FWHM), that can be derived from $(\Sigma, \Delta, \mathrm{K})$ as follows. Let us consider an elongated 2 dimensional Gaussian profile that is resulted by the convolution of a symmetric profile with the matrix

$$\mathbf{s} = \begin{pmatrix} \sigma + \delta & \kappa \\ \kappa & \sigma - \delta \end{pmatrix}. \tag{2.26}$$

It can be shown that such a profile described by (σ, δ, κ) has a covariance of

$$\mathbf{S} = \begin{pmatrix} \sigma + \delta & \kappa \\ \kappa & \sigma - \delta \end{pmatrix}^2, \tag{2.27}$$

i.e. for such profiles, $\mathbf{s}^2 = \mathbf{S}$. Since the FWHM of a Gaussian profile with σ standard deviation is $2\sigma\sqrt{2\log 2} \approx 2.35\,\sigma$, one can obtain the FWHM by calculating the square root of the matrix defined in equation (2.25) and multiply the trace of the root (that is 2σ) by the factor 1.17. Therefore, for nearly circular profiles, the FWHM can be well approximated by $\sim 2.35\sqrt{\Sigma}$.

Finally, the total flux of the object is

$$f = \sum_i (f_i - B), \tag{2.28}$$

and the peak intensity is

$$A = \max_i (f_i - B). \tag{2.29}$$

Analytic models

In order to have a better characterization for the stellar profiles, it is common to fit an analytic model function to the pixels. Such a model has roughly the same set of parameters: background level, flux (or peak intensity), centroid coordinates and shape parameters. The most widely used models are the Gaussian profile (symmetric or elongated) and the Moffat profile. In the characterization of stellar profiles, Lorentz profile and/or Voight profile are not used since these profiles are not integrable in two dimension.

In the cases of undersampled images, we found that the profiles can be well character-ized by the Gaussian profiles, therefore in the practical implementations (see `fistar` and `firandom`, Sec. 2.12.8, Sec. 2.12.7) we focused on these models. Namely, these implementa-tions support three kind of analytic models, both are derivatives of the Gaussian function. The first model is the symmetric Gaussian profile, characterized by five parameters: the background level B, the peak intensity A, the centroid coordinates $\mathbf{x}_0 = (x_0, y_0)$ and the parameter S that is defined as $S = \sigma^{-2}$, where σ is the standard deviation of the profile function. Thus, the model for the flux distribution is

$$f_{\text{sym}}(\mathbf{x}) = B + A \exp\left[-\frac{1}{2}S(\mathbf{x} - \mathbf{x}_0)^2\right]. \tag{2.30}$$

The second implemented model is the elongated Gaussian profile that is characterized by the above five parameters extended with two additional parameters, resulting a flux distribution of

$$f_{\text{elong}}(\mathbf{x}) = B + A \exp\left\{-\frac{1}{2}\left[S(\Delta x^2 + \Delta y^2) + \right.\right. \tag{2.31}$$
$$\left.\left. + D(\Delta x^2 - \Delta y^2) + K(2\Delta x \Delta y)\right]\right\}, \tag{2.32}$$
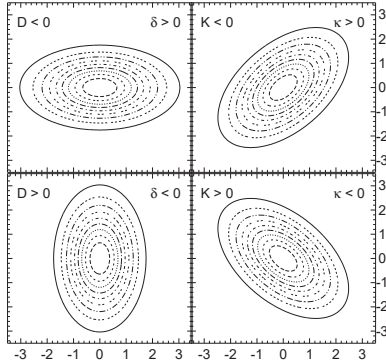
Figure 2.10: Some analytic elongated Gaussian stellar profiles. Each panel shows a contour plot for a profile where the sharpness parameter $S = 1$ and either $|D| = 0.5$ or $|K| = 0.5$. Note that if the Gaussian polynomial coefficients D and/or K are positive, then the respective asymmetric covariance matrix elements Δ and/or K (and the asymmetric convolution parameters δ and/or κ) are negative and vice versa.

where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0$ and D and K are the two additional parameters, that show how the flux deviates from a symmetric distribution. It is easy to show that the (S, D, K) parameters are related to the covariance parameters $(\Sigma, \Delta, \mathrm{K})$ as

$$\begin{pmatrix} S + D & K \\ K & S - D \end{pmatrix} = \begin{pmatrix} \Sigma + \Delta & \mathrm{K} \\ \mathrm{K} & \Sigma - \Delta \end{pmatrix}^{-1}. \tag{2.33}$$

The third model available in the implementations describes a flux distribution that is called "deviated" since the peak intensity is offset from the mean centroid coordinates. Stellar profiles that can only be well characterized by such a flux distribution model are fairly common among images taken with fast focal ratio instruments due to the strong comatic aberration. Such a model function can be built from a Gaussian flux distribution by multiplying the main function by a polynomial:

$$f_{\mathrm{dev}} = B + A \exp\left[-\frac{1}{2} S (\Delta\mathbf{x})^2 \right] \left(1 + \sum_{k,\ell} P_{k\ell} \Delta x^k \Delta y^\ell \right). \tag{2.34}$$

In the summation of equation (2.34), $2 \leq k + \ell \leq M$, where M is the maximal polynomial order and $P_{02} + P_{20}$ is constrained to be 0. Therefore, for $M = 2, 3$ or 4 the above function involves 2, 6 and 11 other parameters in addition to the 5 parameters of the symmetric Gaussian profile. If $M = 2$, the above polynomial is equivalent to the second order expansion of the elongated Gaussian model if $P_{20} - P_{02} = -\frac{1}{2} D$ and $P_{11} = K$. However, for $M = 2$ the peak intensity is not offset from the mean centroid coordinates, therefore in practice $M = 2$ is not used.
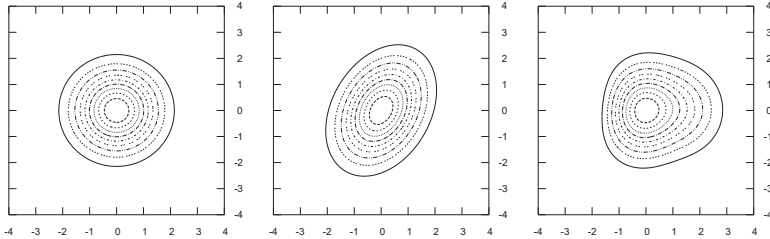
Figure 2.11: Analytic models for stellar profiles. From left to right, the three panels show the contour plots for a symmetric Gaussian profile, for an elongated Gaussian profile and a deviated profile model of $M = 4$. Note that all of the three models have a peak intensity at the coordinate $(0,0)$. In the plots the peak intensity is normalized to unity and the contours show the intensity levels with a step size of 0.1. All of the plotted models have an $S = 1$ parameter while the other parameters (D, K and $P_{k\ell}$) have a value around $\sim 0.1 - 0.2$. Because the choice of $S = 1$, all of the models plotted here has a FWHM of nearly 2.35.

All of the model functions discussed above are nonlinear in the centroid coordinates $\mathbf{x}_0$ and the shape parameters S, D, K or $P_{k\ell}$. Therefore, in a parameter fit, one can use the Levenberg-Marquardt algorithm (Press et al., 1992) since the parametric derivatives of the model functions can easily be calculated and using the parameters of the statistical mean coordinates and standard deviations as initial values yields a good convergence. Moreover, if the iterations of the Levenberg-Marquardt algorithm fail to converge, it is a good indicator to discard the source from our list since it is more likely to be a hot pixel or a structure caused by cosmic ray event[12].

In practice of HATNet and follow-up data reduction, we are using the above models as follows. In real-time applications, for example when the guiding correction is based on the astrometric solution, the derivation of profile centroid coordinates is based on the weighted statistical mean of the pixel coordinates (and in this case, we are not even interested in the shape parameters, just in the centroid coordinates). If more precise coordinates are needed, for example when one has to derive the individual astrometric solutions in order to have a list of coordinates for photometry, the symmetric Gaussian or the elongated Gaussian models are used. The elongated model is also used when we characterize the spatial variations of the stellar profiles. This is particularly important when the optics is not adjusted to the detectors: if the optical axis is not perpendicular to the plane of the CCD chip, the spatial variations in the D and K parameters show a linear trend across the image. If the optical axis is set properly, the linear trend disappears[13]. Finally, if we need to have an analytic description for the stellar profiles as precise as possible, it is worth to use the deviated model.

---

[12]Both cosmic ray events and hot pixels are hard to be modelled with these analytic functions.

[13]Moreover, *quadratic* trends in the D or K components may also be there even if the optical axis is aligned properly. In this case, the magnitude of the quadratic trends is proportional to the magnitude of comatic aberration or the focal plane curveture.

2.4.3   Implementation

The algorithms for extracting stars and characterizing stellar profiles have been implemented in a standalone binary program named `fistar`, part of the FI/FIHAT package. All of the analytic models described here are available in the program `firandom` of which main purpose is to generate artificial images. The capabilities of both programs are discussed in Sec. 2.12 in more detail.

2.5   Astrometry

In the context of reduction of astronomical images, astrometry refers to basically two things. First, the role of finding the astrometrical solution is to find the appropriate function that maps the celestial coordinate system to the image frame and vice versa. Second, the complete astrometrical solution for any given image should identify the individual sources (i.e. perform a "cross-matching"), mostly based on a catalog that is assumed to be known in advance.

Theoretically there is no need to have a list from the available sources found on the image and to have a pre-defined image to find. If one can use only the pixel intensity information of the current and a previously analyzed image to determine a relative transformation and supposing an astrometrical solution being obtained for another image, the two mappings can be composed that results the astrometrical transformation for the current one. This kind of transformations are mostly compositions of dilatation, small rotation and shift if the frames have been acquired subsequently by the same instrumentation from the same stellar field. Such attempts of finding the relative transformation based on only the pixel intensities have been made by Thiebaut & Boër (2001).

In this section, a robust and fast algorithm is presented, for performing astrometry and source cross-identification on two dimensional point lists, such as between a catalogue and an astronomical image, or between two images. The method is based on minimal assumptions: the lists can be rotated, magnified and inverted with respect to each other in an arbitrary way. The algorithm is tailored to work efficiently on wide fields with large number of sources and significant non-linear distortions, as long as the distortions can be approximated with linear transformations locally, over the scale-length of the average distance between the points. The procedure is based on symmetric point matching in a newly defined continuous triangle space that consists of triangles generated by an extended Delaunay triangulation.

2.5.1   Introduction

Cross-matching two two-dimensional points lists is a crucial step in astrometry and source identification. The tasks involves finding the appropriate geometrical transformation that

transforms one list into the reference frame of the other, followed by finding the best matching point-pairs.

One of the lists usually contains the pixel coordinates of sources in an astronomical image (e.g. point-like sources, such as stars), while the other list can be either a reference catalog with celestial coordinates, or it can also consist of pixel coordinates that originate from a different source of observation (another image). Throughout this section we denote the reference (list) as $\mathcal{R}$, the image (list) as $\mathcal{I}$, and the function that transforms the reference to the image as $\mathcal{F}_{R \rightarrow I}$.

The difficulty of the problem is that in order to find matching pairs, one needs to know the transformation, and vica versa: to derive the transformation, one needs point-pairs. Furthermore, the lists may not fully overlap in space, and may have only a small fraction of sources in common.

By making simple assumptions on the properties of $\mathcal{F}_{R \rightarrow I}$, however, the problem can be tackled. A very specific case is when there is only a simple translation between the lists, and one can use cross-correlation techniques (see Phillips & Davis, 1995) to find the transformation. We note, that a method proposed by Thiebaut & Boër (2001) uses the whole image information to derive a transformation (translation and magnification).

A more general assumption, typical to astronomical applications, is that $\mathcal{F}_{R \rightarrow I}$ is a similarity transformation (rotation, magnification, inversion, without shear), i.e. $\mathcal{F}_{R \rightarrow I} = \lambda \mathbf{A} \underline{r} + \underline{b}$, where $\mathbf{A}$ is a (non-zero) scalar λ times the orthogonal matrix, $\underline{b}$ is an arbitrary translation, and $\underline{r}$ is the spatial vector of points. Exploiting that geometrical patterns remain similar after the transformation, more general algorithms have been developed that are based on pattern matching (Groth, 1986; Valdes et al., 1995). The idea is that the initial transformation is found by the aid of a specific set of patterns that are generated from a subset of the points on both $\mathcal{R}$ and $\mathcal{I}$. For example, the subset can be that of the brightest sources, and the patterns can be triangles. With the knowledge of this initial transformation, more points can be cross-matched, and the transformation between the lists can be iteratively refined. Some of these methods are implemented as an IRAF task in `immatch` (Phillips & Davis, 1995).

The above pattern matching methods perform well as long as the dominant term in the transformation is linear, such as for astrometry of narrow field-of-view (FOV) images, and as long as the number of sources is small (because of the large number of patterns that can be generated – see later). In the past decade of astronomy, with the development of large format CCD cameras or mosaic imagers, many wide-field surveys appeared, such as those looking for transient events (e.g. ROTSE — Akerlof et al. 2000), transiting planets (Chapter 1), or all-sky variability (e.g. ASAS – Pojmanski 1997). There are non-negligible, higher order distortion terms in the astrometric solution that are due to, for instance, the

projection of celestial to pixel coordinates and the properties of the fast focal ratio optical systems. Furthermore, these images may contain $\sim 10^5$ sources, and pattern matching is non-trivial.

The presented algorithm is based on, and is a generalization of the above pattern matching algorithms. It is very fast, and works robustly for wide-field imaging with minimal assumptions. Namely, we assume that: i) the distortions are non-negligible, but small compared to the linear term, ii) there exists a smooth transformation between the reference and image points, iii) the point lists have a considerable number of sources in common, and iv) the transformation is locally invertible.

This section has the following parts. First we describe symmetrical point matching in Sec. 2.5.2 before we go on to the discussion of finding the transformation (Sec. 2.5.3). The software implementation and its performance on a large and inhomogeneous dataset is demonstrated in Sec. 2.5.4.

2.5.2 Symmetric point matching

First, let us assume that $\mathcal{F}_{R \to I}$ is known. To find point-pairs between $\mathcal{R}$ and $\mathcal{I}$ one should first transform the reference points to the reference frame of the image: $\mathcal{R}' = \mathcal{F}_{R \to I}(\mathcal{R})$. Now it is possible to perform a simple symmetric point matching between $\mathcal{R}'$ and $\mathcal{I}$. One point ($R_1 \in \mathcal{R}'$) from the first and one point ($I_1 \in \mathcal{I}$) from the second set are treated as a pair if the closest point to R_1 is I_1 *and* the closest point to I_1 is R_1. This requirement is symmetric by definition and excludes such cases when e.g. the closest point to R_1 is I_1, but there exists an R_2 that is even closer to I_1, etc.

In one dimension, finding the point of a given list nearest to a specific point (x) can be implemented as a binary search. Let us assume that the point list with N points is ordered in ascending order. This has to be done only once, at the beginning, and using the quicksort algorithm, for example, the required time scales on average as $\mathcal{O}(N \log N)$. Then x is compared to the median of the list: if it is less than the median, the search can be continued recursively in the first $N/2$ points, if it is greater than the median, the second $N/2$ half is used. At the end only one comparison is needed to find out whether x is closer to its left or right neighbor, so in total $1 + \log_2(N)$ comparisons are needed, which is an $\mathcal{O}(\log N)$ function of N. Thus, the total time including the initial sorting also goes as $\mathcal{O}(N \log N)$.

As regards a two dimensional list, let us assume again, that the points are ordered in ascending order by their x coordinates (initial sorting $\sim \mathcal{O}(N \log N)$), and they are spread uniformly in a square of unit area. Finding the nearest point in x coordinate also requires $\mathcal{O}(\log N)$ comparisons, however, the point found presumably is not the nearest in Euclidean distance. The expectation value of the distance between two points is $1/\sqrt{N}$, and thus we have to compare points within a strip with this width and unity height, meaning $\mathcal{O}(\sqrt{N})$

comparisons. Therefore, the total time required by a symmetric point matching between two catalogs in two dimensions requires $\mathcal{O}(N^{3/2} \log N)$ time.

We note that finding the closest point within a given set of points is also known as nearest neighbor problem (for a summary see Gionis, 2002, and references therein). It is possible to reduce the computation time in 2 dimensions to $\mathcal{O}(N \log N)$ by the aid of Voronoi diagrams and Voronoi cells, but we have not implemented such an algorithm in our matching codes.

2.5.3 Finding the transformation

Let us go back to finding the transformation between $\mathcal{R}$ and $\mathcal{I}$. The first, and most crucial step of the algorithm is to find an initial "guess" $\mathcal{F}_{R \to I}^{(1)}$ for the transformation based on a variant of triangle matching. Using $\mathcal{F}_{R \to I}^{(1)}$, $\mathcal{R}$ is transformed to $\mathcal{I}$, symmetric point-matching is done, and the paired coordinates are used to further refine the transformation (leading to $\mathcal{F}_{R \to I}^{(i)}$ in iteration i), and increase the number of matched points iteratively. A major part of this section is devoted to finding the initial transformation.

Triangle matching

It was proposed earlier by Groth (1986), Stetson (1989) and (see Valdes et al., 1995) to use triangle matching for the initial "guess" of the transformation. The total number of triangles that can be formed using N points is $N(N - 1)(N - 2)/6$, an $\mathcal{O}(N^3)$ function of N. As this can be an overwhelming number, one can resort to using a subset of the points for the vertices of the triangles to be generated. One can also limit the parameters of the triangles, such as exclude elongated or large (small) triangles.

As triangles are uniquely defined by three parameters, for example the length of the three sides, these parameters (or their appropriate combinations) naturally span a 3-dimensional triangle space. Because our assumption is that $\mathcal{F}_{R \to I}$ is dominated by the linear term, to first order approximation there is a single scalar magnification between $\mathcal{R}$ and $\mathcal{I}$ (besides the rotation, chirality and translation). It is possible to reduce the triangle space to a normalized, two-dimensional triangle space $((T_x, T_y) \in T)$, whereby the original size information is lost. Similar triangles (with or without taking into account a possible flip) can be represented by the same points in this space, alleviating triangle matching between $\mathcal{R}$ and $\mathcal{I}$.

Triangle spaces

There are multiple ways of deriving normalized triangle spaces. One can define a "mixed" normalized triangle space $T^{(\mathrm{mix})}$, where the coordinates are insensitive to inversion between the original coordinate lists, i.e. all similar triangles are represented by the same point

irrespective of their chirality (Valdes et al., 1995):

$$T_x^{(\text{mix})} = p/a, \tag{2.35}$$

$$T_y^{(\text{mix})} = q/a, \tag{2.36}$$

where a, p and q are the sides of the triangle in descending order. Triangles in this space are shown on the left panel of Fig. 2.12. Coordinates in the mixed triangle space are continuous functions of the sides (and therefore of the spatial coordinates of the vertices of the original triangle) but the orientation information is lost. Because we assumed that $\mathcal{F}_{R \to I}$ is smooth and bijective, no local inversions and flips can occur. In other words, $\mathcal{R}$ and $\mathcal{I}$ are either flipped or not with respect to each other, but chirality does not have a spatial dependence, and there are no "local spots" that are mirrored. Therefore, using mixed triangle space coordinates can yield false triangle matchings that can lead to an inaccurate initial transformation, or the match may even fail. Thus, for large sets of points and triangles it is more reliable to fix the orientation of the transformation. For example, first assume the coordinates are not flipped, perform a triangle match, and if this match is unsatisfactory, then repeat the fit with flipped triangles.

This leads to the definition of an alternative, "chiral" triangle space:

$$T_x^{(\text{chir})} = b/a, \tag{2.37}$$

$$T_y^{(\text{chir})} = c/a, \tag{2.38}$$

where a, b and c are the sides in counter-clockwise order and a is the longest side. In this space similar triangles with different orientations have different coordinates. The shortcoming of $T^{(\text{chir})}$ is that it is not continuous: a small perturbation of an isosceles triangle can result in a new coordinate that is at the upper rightmost edge of the triangle space.

In the following, we show that it is possible to define a parametrization that is both continuous and preserves chirality. Flip the chiral triangle space in the right panel of Fig. 2.12 along the $T_x + T_y = 1$ line. This transformation moves the equilateral triangle into the origin. Following this, apply radial magnification of the whole space to move the $T_x + T_y = 1$ line to the $T_x^2 + T_y^2 = 1$ arc (the magnification factor is not constant: 1 along the direction of x and y-axis and $\sqrt{2}$ along the $T_x = T_y$ line). Finally, apply an azimuthal slew by a factor of 4 to identify the $T_y = 0, T_x > 0$ and $T_x = 0, T_y > 0$ edges of the space. To be more specific, let us denote the sides as in $T^{(\text{chir})}$: a, b and c in counter-clockwise order where a is the longest, and define

$$\alpha = 1 - b/a, \tag{2.39}$$

$$\beta = 1 - c/a. \tag{2.40}$$

Using these values, it is easy to prove that by using the definitions of the following variables:

$$x_1 = \frac{\alpha(\alpha + \beta)}{\sqrt{\alpha^2 + \beta^2}}, \tag{2.41}$$
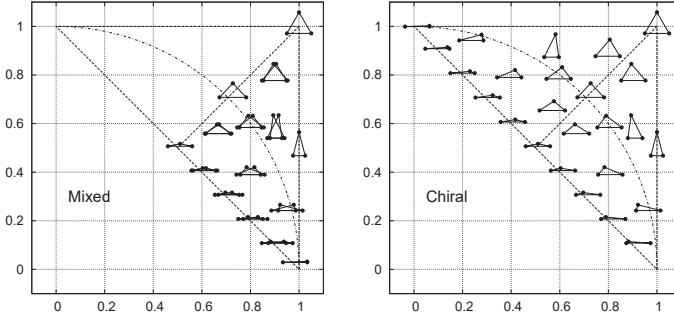
Figure 2.12: The position of triangles in the mixed and the chiral triangle spaces. The exact position of a given triangle is represented by its center of gravity. Note that in the mixed triangle space some triangles with identical side ratios but different orientation overlap. The dashed line shows the boundaries of the triangle space. The dotted-dashed line represents the right triangles and separates obtuse and acute ones.

$$y_1 = \frac{\beta(\alpha + \beta)}{\sqrt{\alpha^2 + \beta^2}}, \tag{2.42}$$

$$x_2 = x_1^2 - y_1^2, \tag{2.43}$$

$$y_2 = 2x_1 y_1, \tag{2.44}$$

one can define the triangle space coordinates as:

$$T_x^{(\text{cont})} = \frac{x_2^2 - y_2^2}{(\alpha + \beta)^3} = \frac{(\alpha + \beta)\left(\alpha^4 - 6\alpha^2\beta^2 + \beta^4\right)}{(\alpha^2 + \beta^2)^2}, \tag{2.45}$$

$$T_y^{(\text{cont})} = \frac{2x_2 y_2}{(\alpha + \beta)^3} = \frac{4(\alpha + \beta)\alpha\beta(\alpha^2 - \beta^2)}{(\alpha^2 + \beta^2)^2}. \tag{2.46}$$

The above defined $T^{(\text{cont})}$ continuous triangle space has many advantages. It is a continuous function of the sides for all non-singular triangles, and also preserves chirality information. Furthermore, it spans a larger area, and misidentification of triangles (that may be very densely packed) is decreased. Some triangles in this space are shown in Fig. 2.13.

Optimal triangle sets

As it was mentioned before, the total number of triangles that can be formed from N points is $\approx N^3/6$. Wide-field images typically contain $\mathcal{O}(10^4)$ points or more, and the total number of triangles that can be generated – a complete triangle list – is unpractical for the following reasons. First, storing and handling such a large number of triangles with typical computers is inconvenient. To give an example, a full triangulation of 10,000 points yields $\sim 1.7 \times 10^{11}$ triangles.

Second, this complete triangle list includes many triangles that are not optimal to use. For example large triangles can be significantly distorted in $\mathcal{I}$ with respect to $\mathcal{R}$, and thus are
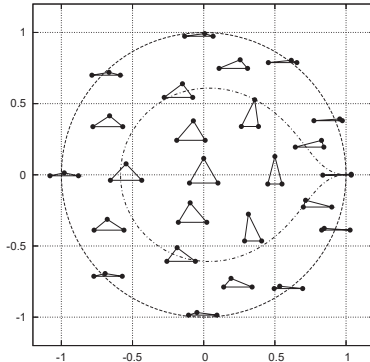
Figure 2.13: Triangles in the continuous triangle space as defined by Eqs. 2.45–2.46. We show the same triangles as earlier, in Fig. 2.12, for the $T^{(\mathrm{mix})}$ and $T^{(\mathrm{chir})}$ triangle spaces. Equilateral triangles are centered in the origin. The dotted-dashed line refers to the right triangles, and divides the space to acute (inside) and obtuse (outside) triangles. Isosceles triangles are placed on the x-axis (where $T_y^{(\mathrm{cont})} = 0$).

represented by substantially different coordinates in the triangle space. The size of optimal triangles is governed by two factors: the distortion of large triangles, and the uncertainty of triangle parameters for small triangles that are comparable in size to the astrometric errors of the vertices.

To make an estimate of the optimal size for triangles, let us denote the characteristic size of the image by D, the astrometric error by δ, and the size of a selected triangle as L. For the sake of simplicity, let us ignore the distortion effects of a complex optical assembly, and estimate the distortion factor f_d in a wide field imager as the difference between the orthographic and gnomonic projections (see Calabretta & Greisen, 2002):

$$f_d \approx |(\sin(d) - \tan(d))/d| \approx |1 - \cos(d)|, \tag{2.47}$$

where d is the radial distance as measured from the center of the field. For the HATNet frames ($d = D \approx 6°$ to the corners) this estimate yields $f_d \approx 0.005$. The distortion effects yield an error of $f_d L/D$ in the triangle space – the bigger the triangle, the more significant the distortion. For the same triangle, astrometric errors cause an uncertainty of δ/L in the triangle space that decreases with increasing L. Making the two errors equal,

$$\frac{f_d \cdot L}{D} = \frac{\delta}{L}, \tag{2.48}$$

an optimal triangle size can be estimated by

$$L_{\mathrm{opt}} = \sqrt{\frac{\delta \cdot D}{f_d}}. \tag{2.49}$$
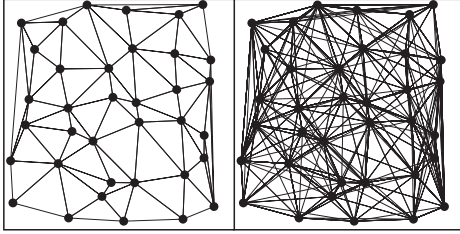
Figure 2.14: Triangulations of some randomly distributed points: the left panel shows the Delaunay triangulation (60 triangles in total) the right panel exhibits the $\ell = 1$ extended triangulation (312 triangles) of the same point set.

In our case $d = 2048$ pixels (or $6°$), $f_d = 0.005$ and the centroid uncertainty for an $I = 11$ star is $\delta = 0.01$, so the optimal size of the triangles is $L_{\text{opt}} \approx 60 - 70$ pixels.

Third, dealing with many triangles may result in a triangle space that is over-saturated by the large number of points, and may yield unexpected matchings of triangles. In all definitions of the previous subsection, the area of the triangle space is approximately unity. Having triangles with an error of σ in triangle space and assuming them to have a uniform distribution, allowing a 3σ spacing between them, and assuming $\sigma = \delta/L_{\text{opt}}$, the number of triangles is delimited to:

$$T_{\text{max}} \approx \frac{1}{(3\sigma)^2} \approx \frac{1}{9}\left(\frac{L}{\delta}\right)^2 = \frac{D}{9 f_d \delta}. \tag{2.50}$$

In our case (see values of D, f_d and δ above) the former equation yields $T_{\text{opt}} \approx 2 \times 10^6$ triangles. Note that this is 5 orders of magnitude smaller than a complete triangulation ($\mathcal{O}(10^{11})$).

The extended Delaunay triangulation

Delaunay triangulation (see Shewchuk, 1996) is a fast and robust way of generating a triangle mesh on a point-set. The Delaunay triangles are disjoint triangles where the circumcircle of any triangle contains no other points from any other triangle. This is also equivalent to the most efficient exclusion of distorted triangles in a local triangulation. For a visual example of a Delaunay triangulation of a random set of points, see the left panel of Fig. 2.14.

Following Euler's theorem (also known as the polyhedron formula), one can calculate the number of triangles in a Delaunay triangulation of N points:

$$T_D = 2N - 2 - C, \tag{2.51}$$

where C is the number of edges on the convex hull of the point set. For large values of N, T_D can be estimated as $2N$, as $2+C$ is negligible. Therefore, if we select a subset of points (from

$\mathcal{R}$ or $\mathcal{I}$) where neighboring ones have a distance of L_{opt}, we get a Delaunay triangulation with approximately $2D^2/L_{\text{opt}}^2$ triangles. The D, δ and f_d values for HAT images correspond to ≈ 6000 triangles, i.e. 3000 points. In our experience, this yields very fast matching, but it is not robust enough for general use, because of the following reasons.

Delaunay triangulation is very sensitive for removing a point from the star list. According to the polyhedron formula, on the average, each point has 6 neighboring points and belongs to 6 triangles. Because of observational effects or unexpected events, the number of points fluctuates in the list. To mention a few examples, it is customary to build up $\mathcal{I}$ from the brightest stars in an image, but stars may get saturated or fall on bad columns, and thus disappear from the list. Star detection algorithms may find sources depending on the changing full-width at half maximum (FWHM) of the frames. Transients, variable stars or minor planets can lead to additional sources on occasions. In general, if one point is removed, 6 Delaunay triangles are destroyed and 4 new ones are formed that are totally disjoint from the 6 original ones (and therefore they are represented by substantially different points in the triangle space). Removing one third of the generating points might completely change the triangulation[14].

Second, and more important, there is no guarantee that the spatial *density* of points in $\mathcal{R}$ and $\mathcal{I}$ is similar. For example, the reference catalog is retrieved for stars with different magnitude limits than those found on the image. If the number of points in common in $\mathcal{R}$ and $\mathcal{I}$ is only a small fraction of the total number of points, the triangulation on the reference and image has no common triangles. Third, the number of the triangles with Delaunay triangulation (T_{D}) is definitely smaller than T_{opt}; i.e. the triangle space could support more triangles without much confusion.

Therefore, it is beneficial to extend the Delaunay triangulation. A natural way of extension can be made as follows. Define a level ℓ and for any given point (P) select all points from the point set of N points that can be connected to P via maximum ℓ edges of the Delaunay triangulation. Following this, one can generate the *full* triangulation of this set and append the new triangles to the whole triangle set. This procedure can be repeated for all points in the point set at fixed ℓ. For self-consistence, the $\ell = 0$ case is defined as the Delaunay triangulation itself. If all points have 6 neighbors, the number of "extended" triangles *per data point* is:

$$T_\ell = (3\ell^2 + 3\ell + 1)(3\ell^2 + 3\ell)(3\ell^2 + 3\ell - 1)/6 \qquad (2.52)$$

for $\ell > 0$, i.e. this extension introduces $\mathcal{O}(\ell^6)$ new triangles. Because some of the extended triangles are repetitions of other triangles from the original Delaunay triangulation and from the extensions of another points, the final dependence only goes as $\mathcal{O}(T_{\text{D}}\ell^2)$. We note that

---

[14]Imagine a honey-bee cell structure where all central points of the hexagons are added or removed: these two construction generates disjoint Delaunay triangulations.

our software implementation is slightly different, and the expansion requires $\mathcal{O}(N\ell^2)$ time and automatically results in a triangle set where each triangle is unique. To give an example, for $N = 10,000$ points the Delaunay triangulation gives $20,000$ triangles, the $\ell = 1$ extended triangulation gives $\sim 115,000$ triangles, $\ell = 2$ some $\sim 347,000$ triangles, $\ell = 3$ $875,000$ and $\ell = 4 \sim 1,841,000$ triangles, respectively. The extended triangulation is not only advantageous because of more triangles, and better chance for matching, but also, there is a bigger variety in size that enhances matching if the input and reference lists have different spatial density.

Matching the triangles in triangle space

If the triangle sets for both the reference and the input list are known, the triangles can be matched in the normalized triangle space (where they are represented by two dimensional points) using the symmetric point matching as described in Sec. 2.5.2.

In the next step we create a $N_R \times N_I$ "vote" matrix V, where N_R and N_I are the number of points in the reference and input lists that were used to generate the triangulations, respectively. The elements of this matrix have an initial value of 0. Each matched triangle corresponds to 3 points in the reference list (identified by r_1, r_2, r_3) and 3 points in the input list (i_1, i_2 and i_3). Knowing these indices, the matrix elements $V_{r_1 i_1}$, $V_{r_2 i_2}$ and $V_{r_3 i_3}$ are incremented. The magnitude of this increment (the *vote*) can depend on the distances of the matching triangles in the triangle space: the closer they are, the higher votes these points get. In our implementation, if N_T triangles are matched in total, the closest pair gets N_T votes, the second closest pair gets $N_T - 1$ votes, and so on.

Having built up the vote matrix, we select the greatest elements of this matrix, and the appropriate points referring to these row and column indices are considered as matched sources. We note that not all of the positive matrix elements are selected, because elements with smaller votes are likely to be due to misidentifications. We found that in practice the upper 40% of the matrix elements yield a robust match.

The unitarity of the transformations

If an initial set of the possible point-pairs are known from triangle-matching, one can fit a smooth function (e.g. a polynomial) that transforms the reference set to the input points. Our assumption was that the dominant term in our transformation is the similarity transformation, which implies that the homogeneous linear part of it should be *almost* unitarity operator[15]. After the transformation is determined, it is useful to measure how much we

---

[15]Here $\mathbf{A}\mathbf{A}^+ = I$, where $\mathbf{A}^+$ is the adjoint of $\mathbf{A}$ and $\mathbf{I}$ is the identity, i.e. $\mathbf{A}$ is an orthogonal transformation with possible inversion and magnification.

diverge from this assumption. As mentioned earlier (Sec. 2.5.1), similarity transformations can be written as

$$\underline{r}' = \lambda \mathbf{A}\underline{r} + \underline{b} \equiv \lambda \begin{pmatrix} a & c \\ b & d \end{pmatrix} \underline{r} + \underline{b}, \tag{2.53}$$

where $\lambda \neq 0$, and the a, b, c, d matrix components are the sine and cosine of a given rotational angle, i.e. $a = d$ and $b = -c$.

If we separate the homogeneous linear part of the transformation, as described by a matrix similar to that in equation (2.53), it is a combination of rotation and dilation with possible inversion if $|a| \approx |d|$ and $|c| \approx |b|$. We can define the unitarity of a matrix as:

$$\Lambda^2 := \frac{(a \mp d)^2 + (b \pm c)^2}{a^2 + b^2 + c^2 + d^2}, \tag{2.54}$$

where the $\pm$ indicates the definition for regular and inverting transformations, respectively. For a combination of rotation and dilation, Λ is zero, for a distorted transformation $\Lambda \approx f_d \ll 1$.

The Λ unitarity gives a good measure of how well the initial transformation was determined. It happens occasionally that the transformation is erroneous, and in our experience, in these cases Λ is not just larger than the expectational value of f_d, but it is ≈ 1. This enables fine-tuning of the algorithm, such as changing chirality of the triangle space, or adding further iterations till satisfactory Λ is reached.

Point matching in practice

In practice, matching points between the $\mathcal{R}$ reference and $\mathcal{I}$ image goes as the following:

1. Generate two triangle sets T_R and T_I on $\mathcal{R}$ and $\mathcal{I}$, respectively:

    (a) In the first iteration, generate only Delaunay triangles.

    (b) Later, if necessary, extended triangulation can be generated with increasing levels of ℓ.

2. Match these two triangle sets in the triangle space using symmetric point matching.

3. Select some possible point-pairs using a vote-algorithm (yielding N_0 pairs).

4. Derive the initial smooth transformation $\mathcal{F}_{R \to I}^{(1)}$ using a least-squares fit.

    (a) Check the unitarity of $\mathcal{F}_{R \to I}^{(1)}$.

    (b) If it is greater than a given threshold ($\mathcal{O}(f_d)$), increase ℓ and go to step (i)/(b). If the unitarity is less than this threshold, proceed to step 5.

(c) If we reached the maximal allowed ℓ, try the procedure with triangles that are flipped with respect to each other between the image and reference, i.e. switch chirality of the $T^{(\text{cont})}$ triangle space.

5. Transform $\mathcal{R}$ using this initial transformation to the reference frame of the image $(\mathcal{R}' = \mathcal{F}_{R \to I}^{(1)}(\mathcal{R}))$.

6. Perform a symmetric point matching between $\mathcal{R}'$ and $\mathcal{I}$ (yielding $N_1 > N_0$ pairs).

7. Refine the transformation based on the greater number of pairs, yielding transformation $\mathcal{F}_{R \to I}^{(i)}$, where i is the iteration number.

8. If necessary, repeat points 5, 6 and 7 iteratively, increase the number of matched points, and refine the transformation.

For most astrometric transformations and distortions it holds that locally they can be approximated with a similarity transformation. At a reasonable density of points on $\mathcal{I}$ and $\mathcal{I}$, the triangles generated by a (possibly extended) Delaunay triangulation are small enough not to be affected by the distortions. The crucial step is the initial triangle matching, and due to the use of local triangles, it proves to be robust procedure. It should be emphasized that $\mathcal{F}_{R \to I}^{(i)}$ can be any smooth transformation, for example an affine transformation with small shear, or polynomial transformation of any reasonable order. The optimal value of the order depends on the magnitude of the distortion. The detailed description of fitting such models and functions can be found in various textbooks (see e.g. Chapter 15. in Press et al., 1992). It is noteworthy that in step 7 one can perform a weighted fit with possible iterative rejection of n-σ outlier points.

2.5.4 Implementation

The coordinate matching and coordinate transforming algorithms are implemented in two stand-alone binary programs as a part of the complete data reduction package. The program named `grmatch` (Sec. 2.12.10) matches point sets, including triangle space generation, triangle matching, symmetric point matching and polynomial fitting, that is steps 1 through 4 in Sec. 2.5.3. The other program, `grtrans` (Sec. 2.12.9), transforms coordinate lists using the transformation coefficients that are output by `grmatch`. The `grtrans` code is also capable of fitting a general polynomial transformation between point-pair lists if they are paired or matched manually or by an external software. We should note that in the case of degeneracy, e.g. when all points are on a perfect lattice, the match fails.

By combining `grmatch` and `grtrans`, one can easily derive the World Coordinate System (WCS) information for a FITS data file. Output of WCS keywords is now fully implemented
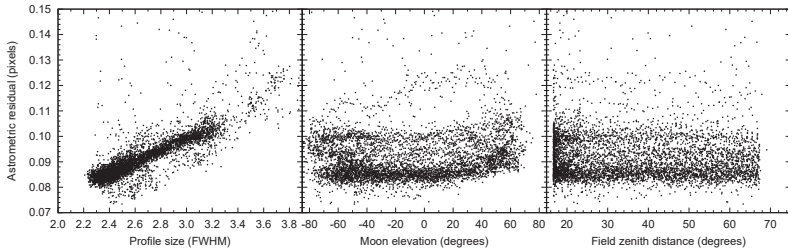
Figure 2.15: Astrometric residuals as the function of observational conditions. The left panel shows the strong correlation between the stellar profile sizes (FWHM $\approx 2.35/\sqrt{S}$): the sharper the stars are on the image the smaller the astrometric residual is. The middle panel shows the astrometric residuals as the elevation of the Moon. Obviously, if the Moon is below the horizon, the residuals are independent from this "negative elevation", however, if the Moon is above the horizon, the effect of the stronger background illumination can be seen well: as the elevation of the Moon increases, the residuals do also become larger. The right panel shows the residuals as the function of the field elevation. No correlation between these can be seen.

in `grtrans`, following the conventions of the package `wcstools`[16] (see Mink, 2002). Such information is very useful for manual analysis with well-known FITS viewers (e.g. `ds9`, see Joye & Mandel, 2003). For a more detailed description of WCS see Calabretta & Greisen (2002) and on the representation of distortions see Shupe (2005).

2.6 Registering images

In order to have data ready for image subtraction, the images themselves have to be transformed to the same reference system (i.e. the images have to be *registered*). This transformation is a continuous mapping between the reference coordinate system and the system of each of the individual images. In practice, all of the frames are taken by the same instrument so this transformation is always nearly identity, affected only by slight rotation, shift and small distortions (for instance due to differential refraction as a given field of the sky is observed at different air masses or small dilations may occur due to the change of focus). In principle, the whole registration process should comply with the following issues. First, the relative transformations are expected to be as small as possible. Since dilations are negligible, the combination of rotation and shift can be described by an affine linear transformation whos determinant is 1. However, the distortions resulted by differential refraction require higher order transformations to be described properly. Second, the flux transformation must preserve brightnesses of the sources. Namely, any area on the image referenced by the same absolute (e.g. celestial) coordinates must contain *exactly* the same amount of flux before and after the transformation is done. Third, composition of the geometric transformations must

---

[16]http://tdc-www.harvard.edu/wcstools/

be as "commutative" as possible with subsequent image transformations. Namely, having two $\mathbb{R}^2 \to \mathbb{R}^2$ mappings, e.g. f and g, and we denote the transformed version of image I by $T_f[I]$, we want to keep $\|T_{f \circ g}[I] - T_f[T_g[I]]\|$ as small as possible. Here "small" means that the difference between the images $T_{f \circ g}[I]$ and $T_f[T_g[I]]$ should be comparable with the overall noise level. In this section the details of this image registration process is discussed.

2.6.1 Choosing a reference image

In practice, the reference image is chosen to be a "nice" image, with high signal-to-noise ratio and therefore with small astrometric residual. Since the signal-to-noise ratio is affected by both the background noise and the fluxes of the individual stars, images taken near culmination, after astronomic twilight and when the Moon is below the horizon are a proper choice in most of the cases. Moreover, in the case of HATNet, sharper images tend to have smaller astrometric residuals because of the merging of nearby stars is also smaller and the background noise affects less pixels. In the panels of Fig. 2.15, the astrometric residuals are shown as the function of the previously discussed observational conditions. As one can expect, the effect of the image sharpness (characterized by the stellar profile FWHMs) and the Moon elevation definitely influence the astrometric residuals. However, the effect of the field elevation itself is negligible, the variation in the airmass between ~ 1.02 and ~ 2.55 (i.e. $12° \lesssim z \lesssim 67°$) causes no practical fluctuation in the astrometric residuals.

We should note here that the whole process of the image subtraction photometry needs not only a specific image to be an astrometric reference but a couple of images for photometric reference as well. As we will see later on, the selection criteria for convolution reference images are roughly the same as for an astrometric reference. Hence, in practice, the astrometric reference image is always one of the convolution reference frames.

2.6.2 Relative transformations

Once the reference frame for registration has been chosen, the appropriate geometric transformations between this frame and the other frames should be derived (prior to the image transformation itself). To derive this geometric transformation, one can proceed using one of the following methods:

- Assuming the absolute astrometrical solutions to be known (i.e. the mappings between the celestial and pixel coordinates), the solution for the reference frame can be composed with the inverse of the solution for the current image.

- Assuming that the sources on both the reference and the current images are extracted and identified with a previously declared external catalog, one can match these iden-
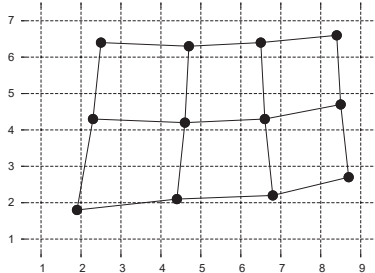
Figure 2.16: In order to perform a spatial image transformation with exact flux conserving, the intensity level of the original image should be integrated on the quadrilaterals defined by the mapping function. Each quadrilateral is the projection of one of the pixels in the target image while the dots represent the projections of the pixel corners. The above image shows the pixel grid of the original image and the grid of quadrilaterals for a transformation that shrinks the image by a factor of nearly two.

tifier – pixel coordinate lists and fit a geometric transformation involving the matched coordinate pairs.

- If any kind of astrometric information – neither absolute solution nor source identification – is not known in advance, one can directly employ the triangulation-based point matching algorithm itself, as it was presented earlier (Sec. 2.5).

In practice, the first option is sub-optimal. Since the absolute astrometric transformation has higher order distortions than in a relative transformation, such composition does easily lead to numeric round-off errors. Moreover, the direct composition of two polynomials with an order of 6 (which is needed for a proper astrometric solution, see Sec. 2.1.3, Table 2.1 or Sec. 2.5) yields a polynomial with an order of 12, while a relative transformation between two images needs only a polynomial with a degree of $3 - 4$ (see also Table 2.1). The naive omission of higher order polynomial coefficients does not result the "best fit" and this best fit depends on the domain of the polynomial therefore this polynomial degradation is always an ambiguous step.

Both the second or third option mentioned above are efficient and can be used in practice. The last option, involving the point matching to determine the relative transformation has an advantage: on cloudy images where derivation of the absolute astrometric solution failed, the chance to obtain a successful relative transformation is higher. This is mostly because of both the lack of large-scale distortions and the smaller polynomial degree required for such transformations.

2.6.3   Conserving flux

Even if the spatial image transformation does not significantly shrink or enlarge the image, pixels of the target image usually are not mapped exactly to the pixels of the original image (and vice versa). Therefore, some sort of interpolation is needed between the adjacent pixel values in order to obtain an appropriate transformed image. Since the spatial transformation is followed by the steps of convolution and photometry, exact flux conservation is a crucial issue. If the interpolation is performed naively by multiplying the interpolated pixel values with the Jacobian determinant of the spatial mapping, the exact flux conservation property is not guaranteed at all. It is even more relevant in the cases where the transformation includes definite dilation or shrinking, i.e. the Jacobian determinant significantly differ from unity.

In order to overcome the problem of the flux conserving transformations, we have implemented a method based on analytical integration of surfaces of which are determined by the pixel values. These surfaces are then integrated on the quadrilaterals whose coordinates are derived by mapping the pixel coordinates on the target frame to the system of the original frame. An example is shown in Fig. 2.16, where the transformation includes a shrink factor of nearly two (thus the Jacobian determinant is $\sim 1/4$). In practice, two kind of surfaces are used in the original image. The simplest kind of surface is the two dimensional step function, defined explicitly by the discrete pixel values. Obviously, if the area of the intersections of the quadrilaterals and the pixel squares is derived, the integration is straightforward: it is equivalent with a multiplication of this intersection area by the actual pixel value.

A more sophisticated interpolation surface can be defined as follows. On each pixel, at the position (i, j), we define a biquadratical function of the fractional pixel coordinates $(\delta x, \delta y)$, namely

$$f^{ij}(x, y) = \sum_{k=0}^{2} \sum_{\ell=0}^{2} C_{k\ell}^{ij} \delta x^k \delta y^\ell. \qquad (2.55)$$

For each pixel, we define nine coefficients, $C_{k\ell}^{ij}$. We derive these coefficients by both constraining the integral of the surface at the pixel to be equal to the pixel value itself, i.e.

$$\int_0^1 \int_0^1 f^{ij}(\delta x, \delta y)\, \mathrm{d}\delta x\, \mathrm{d}\delta y = P_{ij}, \qquad (2.56)$$

and requiring the joint function $F(x, y)$ describing the surface

$$F(x, y) = f^{[x][y]}(\{x\}, \{y\}) \qquad (2.57)$$

to be continuous (here $[x]$ denotes the integer part of x and $\{x\}$ denotes the fractional part, i.e. $x = [x] + \{x\}$). This continuity is equivalent to

$$f^{[i+1]j}(0, y) = f^{ij}(1, y), \qquad (2.58)$$

$$f^{[i-1]j}(1,y) = f^{ij}(0,y), \qquad (2.59)$$

$$f^{i[j+1]}(x,0) = f^{ij}(x,1), \qquad (2.60)$$

$$f^{i[j-1]}(x,1) = f^{ij}(x,0), \qquad (2.61)$$

for all $0 \leq x, y \leq 1$. Since f is a biquadratical function of the fractional pixel coordinates (x,y), it can be shown that the above four equations imply 8 additional constraints for each pixel. At the boundaries of the image, we can define any feasible boundary condition. For instance, by fixing the partial derivatives $\partial F/\partial x$ and $\partial F/\partial y$ of the surface $F(x,y)$ to be zero at the left/right and the lower/upper edge of the image, respectively. It can be shown that the integral property of equation (2.56), the continuity constrained by equations (2.58)-(2.61) and the boundary conditions define an unique solution for the $C_{k\ell}^{ij}$ coefficients. This solution exists for arbitrary values of the P_{ij} pixel intensities (note that the complete problem of obtaining the $C_{k\ell}^{ij}$ is a system of linear equations). Since the integrals of the $F(x,y)$ surface on the quadrilaterals are linear combinations of polynomial integrals, the pixel intensities on interpolated images can be obtained easily, although it is a bit more computationally expensive.

We should note here that if the transformation is a simple shift (i.e. there are not any dilation, rotation and higher order distortions at all), the two, previously discussed interpolation schemes yield the same results as the classic bilinear and bicubic (Press et al., 1992) interpolation.

In practice, during the above interpolation procedure pixels that have been marked to be inappropriate[17] are ignored from the determination of the $C_{k\ell}^{ij}$ coefficients, and any interpolated pixels on the target image inherit the underlying masks of the pixels that intersects their respective quadrilaterals. Pixels on the target frame that are mapped off the original image have a special mask which marks them "outer" ones (see also Sec. 2.3.2). It yields a transparent processing of the images: for instance in the case of photometry, if the aperture falls completely inside the image but intersects one or more pixels having this "outer" mask yields the same photometry quality flag as if the aperture is (partially or completely) off the image. See also Sec. 2.7 or Sec.2.12.13 for additional details.

2.6.4 Implementation

The core algorithms of the interpolations discussed here are implemented in the program `fitrans` (Sec. 2.12.11). This program performs the spatial image transformation, involving both the naive and the integration-based methods and both the bilinear and bicubic/biquadratical interpolations. The transformation itself is the output of the `grmatch` or `grtrans` programs (see also Sec. 2.12.10 and Sec. 2.12.9).

---

[17]For instance, pixels that are saturated or have any other undesired mask.

2.7 Photometry

The main step in a reduction pipeline intended to measure fluxes of objects on the sky is the photometry. All of the steps discussed before are crucial to prepare the image to be ready for photometry. Thus at this stage we should have a properly calibrated and registered[18] image as well as we have to know the positions of the sources of interest. For each source, the CCD photometry process for a single image yields only *raw instrumental* fluxes. In order to estimate the *intrinsic* flux of a target object, ground-based observations use nearby comparison objects with known fluxes. The difference in the raw instrumental fluxes between the target source and the source with known flux is then converted involving smooth transformations to obtain the ratios between the intrinsic flux values. Such smooth transformation might be the identical transformation (this is the simplest of all photometry methods, known as single star comparison photometry) or some higher order transformations for correcting various gradients (mostly in the transparency: due to the large field of view, the airmass and therefore the extinction at the different corners of the image might significantly differ). Even more sophisticated transformations can also be performed in order to correct additional filter- and instrumentation effects yielded by the intrinsic color (and color differences) between the various sources. Corrections can also made in order to transform the brightnesses into standard photometric systems. The latter is known as standard transformation and almost in all cases it requires measurements for standard areas as well (Landolt, 1992). Since for all objects, transparency variations cause flux increase or decrease proportional to the intrinsic flux itself, the transformations mentioned above are done on a logarithmic scale (in practice, magnitude scale). For instance, in the case of single-star comparison photometry, the difference between the intrinsic magnitudes and the raw instrumental magnitudes is constant[19]. In this section some aspects of the raw and instrumental photometric methods are detailed with the exception of topics related to the photometry on convolved and/or subtracted images. As it was mentioned above, the first step of the photometry is the derivation of the raw instrumental magnitudes of the objects or sources of our interest.

2.7.1 Raw instrumental magnitudes

In principle, raw magnitudes are derived from two quantities. First, the total flux of the CCD pixels are determined around the object centroid. The total flux can be determined in three manners:

---

[18]Only if we intend to perform image subtraction based photometry.

[19]To be precise, only if the spectra of the two stars are exactly the same *and* the two objects are close enough to neglect the difference in the atmospheric transparency.

- If a region is assigned to the object of interest, one has to count the total flux of the pixels inside this region. The region is generally defined to be within a fixed distance from the centroid (so-called *aperture*), but in the case of diffuse or non-point sources, more sophisticated methods have to be used to define the boundary of the region. The algorithms implemented in the program SExtractor (Bertin & Arnouts, 1996) focus on photometry of such sources. In the following we are interested only in stars and/or point-like sources.

- If the source profile can be modelled with some kind of analytic function (see Sec 2.4.2) or an empirical model function (e.g. the PSF of the image), one can fit such a model surface to the pixels that are supposed to belong to the object (e.g. to the pixels being inside of a previously defined aperture or one of the isophotes). From the fitted parameters, the integral of the surface is derived, and this integral is then treated as the flux of the object. This method for photometry is known as PSF photometry.

- The previous two methods can be combined as follows. After fitting the model function, the best fit surface is subtracted from the pixel values and aperture photometry is performed on this residual. The flux derived from the residual photometry is then added to the flux derived from the best fit surface parameters yielding the total flux for the given object. It is not necessary that the pixels used for surface fitting are the same as the pixels being inside the aperture.

It should be mentioned here that whatever primary method from these above is used to perform the photometry, estimating the uncertainties should be done carefully.

After the total flux of the object has been estimated, one has to remove the flux contribution of the background. It is essential in the case of aperture photometry, however, if a profile function is fitted to the pixel values, the contribution of the background is added to the model function as an additional free parameter. If the photometric aperture is a circular region, the background is usually defined as a concentric annulus, whose inner radius is larger than the radius of the aperture. If the field is not crowded, the background level is simply the mean or median of the pixel values found in the annulus. On the other hand, if the field is extremely crowded, the determination of the background level might even be impossible. A solution for this issue can be either profile (PSF) fitting or photometry based on differential images (see Sec. 2.9). Note that on highly crowded fields, apertures significantly overlap. One advantage of the profile/PSF model fitting method is the ability to fit adjacent profiles simultaneously.

In practice, additional data are obtained and reported for a single raw instrumental photometry measurement, such as:

- Noise estimations, based on the Poisson statistics of the flux values, the uncertainty
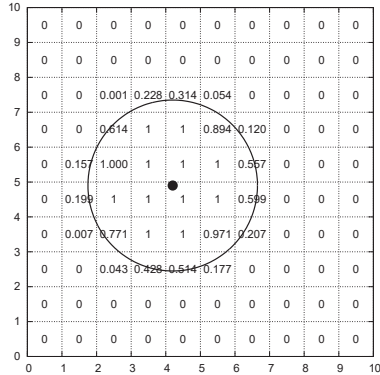
Figure 2.17: Weight matrix for a circular aperture centered at $(x_0, y_0) = (4.2, 4.9)$ and having a radius of $r_0 = 2.45$ pixels. The numbers written in the squares show the area of the intersection of the given square and the circle.

of the background level determination, and optionally scintillation noise can also be estimated (Young, 1967);

- Characteristics of the background: total number of pixels used to derive the background level, number of outlier pixels – such as pixels of nearby stars or cosmics events – rejected from the background determination procedure and so.

- Quality flags, such as various pixel masks happen to fall in the aperture.

2.7.2 Formalism for the aperture photometry

In practice, aperture photometry derives the raw instrumental magnitudes as follows. Let us consider an image I with the pixel intensities $I(x, y) \equiv I_{xy}$ where (x, y) are the respective pixel coordinates. Let us define the weight matrix for the circular aperture centered at (x_0, y_0) and having a radius of r_0 as

$$
\begin{aligned}
A_{xy} &\equiv A_{xy}^{x_0, y_0, r_0} = \\
&= \int_{x_0}^{x_0+1} dx \int_{y_0}^{y_0+1} dy \, \Theta \left[r_0^2 - (x - x_0)^2 - (y - y_0)^2 \right],
\end{aligned}
\tag{2.62}
$$

where $\Theta(\cdot)$ is the Heaviside step function (see also Fig. 2.17). Due to the definition of A_{xy}, it is unity inside the aperture, has some value between 0 and 1 at the boundary (depending on the area of overlap), and it is zero further outside from the aperture centroid. The total

raw instrumental flux f_{total} is then simply derived as

$$f_{\text{total}} = \sum_{x,y} A_{xy} I_{xy}. \tag{2.63}$$

The background level in the annulus having inner and outer radii of r_1 and r_2 respectively, around the centroid (x_0, y_0) can be derived as

$$B = \frac{\sum\limits_{x,y} I_{xy} \left(A_{xy}^{x_0, y_0, r_2} - A_{xy}^{x_0, y_0, r_1} \right)}{r_2^2 - r_1^2}. \tag{2.64}$$

The raw flux of the object in the aperture after the background level removal is

$$f = \sum_{x,y} A_{xy}(I_{xy} - B) = f_{\text{total}} - B r_0^2. \tag{2.65}$$

Albeit this discussion seems to be rather trivial, the same formalism will be used later on in Sec. 2.9 while considering the details of photometry performed on subtracted images.

2.7.3 Magnitude transformations

As it was mentioned earlier, raw magnitude lists on subsequent frames yielded by the photometry have to be transformed to the same reference system in order to have instrumental and/or standard magnitudes for our objects. For a given frame, let us assume to have a list of stars with $m^{(i)}$ raw magnitudes, located at the (x_i, y_i) position on the image. Let us denote the raw magnitudes of these objects on a certain reference frame by $m_0^{(i)}$. For images obtained by small field-of-view instrumentation, the $m^{(i)} - m_0^{(i)}$ difference depends only on the color of the star, due to the wavelength dependence of the atmospheric extinction. For images obtained by larger field-of-view optics, the difference between the instrumental magnitudes depend also on the (x, y) centroid positions due to the gradient in the extinction level throughout the image. In practice, both the spatial and color dependence of the differential magnitudes can be well characterized by polynomials. Such a transformation is quantified as

$$m^{(i)} - m_0^{(i)} = \sum_{c=0}^{N} \left((C^{(i)})^c \sum_{0 \le k+\ell \le N_c} K_{ck\ell} x_i^k y_i^\ell \right), \tag{2.66}$$

where $C^{(i)}$ is some color index (e.g. $V - I$ or $J - K$) of the star, and N and N_c are the maximal polynomial orders in the color and in the spatial coordinates, respectively. The $K_{ck\ell}$ coefficients can be obtained by involving the linear least squares method, if each of the stars are weighted appropriately. The weights assigned to the stars can be derived from both the photon noise and the light curve residuals. In practice, the above mentioned magnitude transformation is done iteratively. First, instrumental magnitude lists for each frame are transformed to the instrumental system of one of the frames. This reference frame is usually

selected from the "best" frames, i.e. that has been obtained at low airmass and good generic atmospheric conditions, has small astrometric residuals and the illumination of the Moon and/or sky background (due to twilight) is the smallest. After each magnitude list have been transformed, light curves are gathered and the individual scatters are derived for each star. The transformation is then repeated while the contribution of each star is weighted by the light curve scatters. This kind of weighting gives lower weight for stars whose scatter have been underestimated (due to unresolved remaining systematics, for instance) or have intrinsic but not known variability. Of course, stars with known variability should be excluded from the fit, including our target stars as well.

2.7.4 Implementation

In the FI/FIHAT package, the above discussed photometric algorithms are implemented as follows. The aperture photometry and the related features – such as background level determination, noise estimations, assignment of quality flags, conversion of fluxes to instrumental magnitudes – are implemented in the program `fiphot` (see Sec. 2.12.13). The point-spread functions are derived by the program `fistar` (Sec. 2.12.8). Moreover, this program is also capable to fit the derived PSFs to the individual detected profiles. Currently, none of these programs deals explicitly with profile fit residuals, however, the output of `fistar` can be used as an input for `firandom` (both for analytical profile models and PSFs) to create model images. Such model images are suitable to subtract from the original images yielding complete residual images. The program `lfit` is another alternative for fitting analytic stellar profile models that are not supported by `fistar`/`firandom`. Magnitude transformations between two frames can also be performed with the program `lfit`. See also Sec. 2.13 and Fig. 2.29 about the practical details about how these programs can be applied for real observations.

2.8 Image convolution and subtraction

In a generic variability survey, such as the HATNet project, we are primarily focusing on the detection and the quantifications of source brightness variations. The idea behind the photometry methods involving image subtraction is to derive the part of the flux that varies from image to image. It is rather easy to see that simple per-pixel arithmetic subtraction is not sufficient to derive the difference between two images. First, the centroid positions of the stars are different for each image. The magnitude of this difference depends on the precision and the systematic variations in the mount tracking, as well as other side effects such as field rotation and the intrinsic differential refraction. However, it is rather easy to overcome this problem by registering the images to the same reference system (Sec. 2.6). Second, background level may vary from image to image. Changes in the background can

be modelled by adding a constant or some slowly varying function to the (convolved) image. Third, the stellar profiles are also vary from frame to frame, due to the variations in the seeing or in the focus. In order to have the smallest residual between two images, one should not only register these to the same reference system but on at least one of the images, the profiles should be transformed to match the profiles of the other image. This profile transformation is performed as a convolution, namely the image R is transformed to R' as

$$R' = B + R \star K, \tag{2.67}$$

where K is the convolution kernel and the operator $(\cdot) \star (\cdot)$ denotes the convolution. For (astronomical) images that are sampled on discrete pixels, the operation of convolution is defined as

$$R'_{xy} = \sum_{-B_K \leq i,j \leq +B_K} R_{(x-i)(y-j)} K_{ij}. \tag{2.68}$$

Here, the convolution kernel K_{ik} is sampled on a grid of $(2B_K + 1) \times (2B_K + 1)$ pixels and I_{xy} refers to the intensity of the pixel at (x, y). If the difference of FWHMs of the image R and R' are small, the kernel can be sampled on a smaller grid. In general, a kernel function with an FWHM of F_K yields a profile FWHM F' on the convolved image of

$$F' \approx \sqrt{F^2 + F_K^2}, \tag{2.69}$$

where F is the FWHM of the profiles on the image R.

Supposing two images, I and R, the main problem of the image convolution and subtraction method is to find the appropriate kernel K with which the image R convolved, the resulting image is nearly identical to I. The first attempt to find this optimal kernel (Tomaney & Crotts, 1996) was based on an inverse Fourier transformation between the two PSFs of the images. Theoretically, inverse Fourier transformation yields the appropriate kernel, however, the practical usage of this method is limited due to the high signal-to-noise ratio that is needed by a Fourier inversion. Kochanski, Tyson & Fischer (1996) attempted to find the kernel K by minimizing the merit function

$$\chi^2_\infty = \sum_{xy} |I_{xy} - (R \star K)_{xy}|. \tag{2.70}$$

This minimization yields a non-linear equation for the kernel K and therefore it is not computationally efficient. The most cited algorithm related to image subtraction was given by Alard & Lupton (1998). In this work, an additional term was added to the convolution transformation, which allows to fit not only the convolution transformation but the background variations:

$$I = B + R \star K. \tag{2.71}$$

The basic idea of Alard & Lupton (1998) was to minimize the function

$$\chi^2 = \sum_{xy} (I_{xy} - [B_{xy} + (R \star K)_{xy}])^2 \tag{2.72}$$

and search the kernel solution K in the form of

$$K = \sum_i C_i K^{(i)}. \tag{2.73}$$

In their work, the kernels K^i were two dimensional Gaussian functions with variable FWHMs multiplied by polynomials. Assuming the background variations to be constant, i.e. $B_{xy} \equiv B$, minimizing equation (2.72) yields a linear set of equations for the parameters B and C_i, thus its solution is straightforward (and efficient). Shortly after, Alard (2000) gave a more sophisticated method that allows the kernel parameters as well as the background level to vary across the image:

$$I_{xy} = B(x, y) + [R \star K(x, y)]_{xy}. \tag{2.74}$$

Both the background variations and the kernel coefficients were searched as a polynomial function of the pixel coordinates, namely

$$B(x, y) = \sum_{0 \leq k+\ell \leq N_{\mathrm{bg}}} B_{k\ell} x^k y^\ell \tag{2.75}$$

and

$$K(x, y) = \sum_i \sum_{0 \leq k+\ell \leq N_{\mathrm{K}}^{(i)}} C_{ik\ell} K^{(i)} x^k y^\ell. \tag{2.76}$$

It is easy to show that finding the optimal $B_{k\ell}$ and $C_{ik\ell}$ coefficients still requires only linear least squares minimization. Alard & Lupton (1998) also discuss how the individual pixels used in the fit must be weighted by the Poisson noise level in order to have a consistent result. Recently, Bramich (2008) searched the optimal kernel K by assuming an alternate set of kernel base functions $K^{(i)}$, involving discrete kernels instead of Gaussian functions. These discrete kernels are defined as

$$K^{(u,v)} = \delta^{(uv)}, \tag{2.77}$$

where

$$(\delta^{(uv)})_{xy} = \begin{cases} 1 & \text{if } u = x \text{ and } v = y, \\ 0 & \text{otherwise.} \end{cases} \tag{2.78}$$

The total number of base kernels is then $N_{\mathrm{kernels}} = (2B_{\mathrm{K}} + 1)^2$. Yuan & Akerlof (2008) attempted to find the solution K_i, B and K_r of the equation

$$I \star K_i = B + R \star K_r. \tag{2.79}$$

This method is known as *cross-convolution* and works properly in the cases when there is no suitable solution for equation (2.71). For instance, on the image R the profiles have such shape parameters where $K > 0$ and $D = 0$ while on the image I these parameters are $K < 0$ and $D = 0$. The method of cross-convolution has a disadvantage, namely if one finds a solution K_i and K_r for equation (2.71), $K_i \star G$ and $K_r \star G$ is also a solution (where G is an arbitrary convolution kernel). Therefore equation (2.71) is degenerated unless additional constraints are introduced (e.g. by minimizing the $\|K_i - K_r\|$ difference simultaneously).

2.8.1 Reference frame

The noise characteristics of the subtracted image is determined by both the reference image R and the target image I. If both images are individual frames, the generic noise level is approximately $\sqrt{2}$ times larger than that of on the individual frames. In order to reduce the noise level on the subtracted frames, the reference image R is created from several individual frames. If the number of such frames is N, the noise level of the subtracted images is $\sqrt{1 + 1/N} \approx 1 + 1/(2N)$ (supposing that both the reference frames and the target image have the same noise level). Thus, a number of $N \approx 20 - 25$ frames are sufficient to increase the noise level on the subtracted image only by a few percent[20].

2.8.2 Registration

As it was seen related to the difficulties of the photometry on undersampled images (Sec. 2.1.1), the interpolation of such images with sharp profiles is likely to yield artifacts, "spline undershoots" and therefore systematic residuals (Fig. 2.4). Since the FWHM of the HATNet frames is too small to clearly remove such residuals, we have used the following sophisticated registration process. First, using the stellar profile parameters and flux estimations yielded by the modelling described in Sec. 2.4.2, a model for the images is created, involving the program `firandom` (Sec. 2.12.7). This image model is then subtracted from the original image, yielding a residual with no sharp structures. The residual image is then transformed to the reference system, simultaneously with the transformation of the centroid coordinates found in the stellar profile parameter list. Using the transformed stellar profile parameters, another model image is created that is added to the transformed residual image. Since the stellar profiles can be well modelled by an analytic function, this way of image registration yields no artifacts on the transformed images, even for highly undersampled profiles. Additionally, we do not have to involve *all* of the stars on the image, only the

---

[20]Strictly speaking, a noisy reference frame implies a *correlated* noise on the subtracted frames since the same image (or its versions derived by convolution) is subtracted from the original frames. Therefore, it is an upper limit for the noise increment in the final light curves. However, the scatter in the convolution parameters also increase the light curve noise, but this cannot be quantified in a simple way.

brighter ones, since for fainter stars the amplitude of spline undershoots are comparable to or less than the noise level.

This kind of transformation is even more relevant during the creation of the reference image R since this image is created by averaging some of the most sharpest images.

2.8.3 Implementation

Those methods discussed above that are based on the technique of linear least squares are implemented in the program `ficonv` (see Sec. 2.12.12). The practical details of the photometry based on the method of image subtraction are explained in Chapter 3, related to the HAT-P-7(b) planetary system.

2.9 Photometry on subtracted images

As it was discussed in the previous section (Sec. 2.8), the method of image convolution and subtraction aids the photometry process by both decreasing the fluctuations in the background level and reducing the influence of the nearby stars on the background area level. A great advantage of the image subtraction method is that it does not need to know about stars (initially), can use all of the pixels and works in extremely crowded images. In the simplest case when both the reference image R and the target image I have exactly the same intensity level and the stellar profiles are nearly the same, the flux of a given star on image I can be obtained by simply adding the reference flux and the flux measured on the residual image.

However, even in the cases where the stellar profiles are nearly the same but the images R and I have different intensity levels (for example, image I was acquired at higher airmass or lower transparency while the reference was chosen to be one of the high signal-to-noise images, acquired at high horizontal altitudes), the photometry on the subtracted images is not as simple as before. Let us consider the following situation. The flux of a given isolated star on the reference image is found to be 1000 ADUs. In the target image, this star has an intrinsic flux decrease of 1%, thus if this image had been acquired under the same conditions as the reference image, the flux of the star would be 990 ADUs. Let us suppose now that due to the low sky transparency, all of the stars have a flux decrease of 50%, thus our star is measured to have a flux of 495 ADUs. The best fit kernel solution that transforms the reference image to the target image is then $K = \frac{1}{2}\delta$. Therefore, the residual flux of the target star would be -5 ADUs. If this residual flux is simply added to the reference flux, the obtained flux is only 995 ADUs, thus the measured flux decrease (the signal itself) is significantly underestimated. Moreover, if the kernel solution of equation (2.71) implies

significant difference between the FWHMs of the stellar profiles in the reference and target image, both the methods of PSF and aperture photometry should be tweaked.

Using the formalism shown in Sec. 2.7.2, aperture photometry on subtracted images can be performed as follows. It is easy to show that for any weight matrix of A_{xy}, the relation

$$\sum_{x,y}(R \star K)_{xy}(A \star K)_{xy} = \sum_{x,y}(R_{xy}A_{xy})\|K\|_1^2 \tag{2.80}$$

is true if the aperture A supports the convolved profile of $R \star K$ and it is a rather good approximation if the aperture has a size that is comparable to the profile FWHM. The norm $\|K\|_p$ is defined as

$$\|K\|_p := \sqrt[p]{\sum_{x,y}|K_{xy}|^p}. \tag{2.81}$$

Moreover, the ratio of the two sides in equation (2.80) is independent from $\|K\|_1$ (even if the aperture A does not support completely the convolved profile on $R \star K$) or in other words, this ratio does not change if K is multiplied by an arbitrary positive constant. Therefore, involving an aperture of A_{xy}, the flux of a source found on the convolved image $C = R \star K$ can be obtained as

$$f_C = \frac{\sum\limits_{x,y} C_{xy}(A \star K)_{xy}}{\|K\|_1^2} \tag{2.82}$$

and this raw flux is independent from the large scale flux level variations that are quantified by $\|K\|$. The total flux f of the source can be derived from the flux on the reference image and the flux of the target image. Since the method of image subtraction tries to find the optimal kernel K, that minimizes $\|I - B - R \star K\|_2$, combining equation (2.82) and equation (2.63) from Sec. 2.7.2, f is obtained as

$$f = \frac{\sum\limits_{x,y} S_{xy}(A \star K)_{xy}}{\|K\|_1^2} + \sum_{x,y}R_{xy}A_{xy}. \tag{2.83}$$

Here S is $I - B - R \star K$, the subtracted image. Of course, one can derive a background level around the target object on the subtracted images, but in most of the cases this background level is zero within reasonable uncertainties. However, it is worth to include such a background correction even on the subtracted images since unpredictable small-scale background variations[21] can occur at any time.

2.10 Trend filtering

Photometric time series might show systematic variations due to various effects. Of course, if a certain star is indeed a variable, the main source of photometric variations should be

---

[21]For instance, variations yielded by thin clouds or scattered light, that cannot be characterized by a function like in equation (2.75)
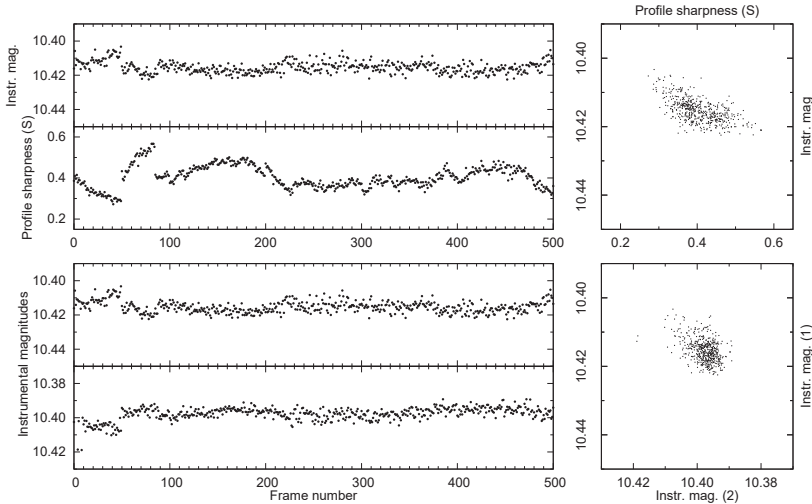
Figure 2.18: Typical examples of trends. The upper panels display the primary concepts of the External Parameter Decorrelation: for a particular star, the lower inset shows the variance in the profile sharpness parameter (S) throughout the night while the upper inset shows the instrumental magnitude. The panel in the upper-right corner shows the distribution of the individual measurements in the $S-$ magnitude parameter space. The correlation between these two parameters can be seen clearly. The lower panels display light curves for two given stars in the same instrumental photometric system. The insets on the left show the two light curves while the plot in the lower-right corner shows the magnitude $-$ magnitude distribution. The correlation between the two magnitudes is quite clear also in this case.

the intrinsic changes in the stellar brightness. However, there are various other effects that yield unexpected trends in the light curves, which still present after the magnitude transformation and even if sophisticated algorithms are involved in the data reduction (such as image subtraction based photometry). The primary reasons for such trends are the following. Observational conditions might vary (even significantly) throughout the night, for instance clouds are blocking the light at some regions of the field, or the background level is increasing due to the twilight or the proximity of the Moon. Additionally, instrumental effects, such as variations in the focal length or drops or increases in the detector temperature can result in various trends. And finally, lack of the proper data reduction is also responsible for such effects. For instance, faults in the calibration process, insufficiently large polynomial orders in the astrometric or magnitude transformations, underestimated or overestimated aperture sizes, badly determined PSFs, inappropriate reference frames; all of these are plausible reasons for unexpected systematic variations. In this section the efforts are summarized intended to reduce the remaining trends in light curves.

The basic concepts of trend removal are the following. First, one can assume that instrumental magnitudes have some remaining dependence on additional quantities that are

also derived during the data reduction. Such external parameters can be the profile shape parameters, centroid coordinates, celestial positions (such as elevation or hour angle of the target field or object), or environmental parameters (external temperature). The dependence on these parameters therefore results in a definite correlation. Assuming some qualitative dependence, these correlations can then be removed, yielding light curves with smaller scatter. The type of the qualitative dependence is related to certain parameters against which the de-correlation is performed (see later on some examples). In general, this method of the External Parameter Decorrelation (EPD; see e.g. Bakos et al., 2007b) yields a linear least squares fit. Second, either if we have no information about all of the external parameters or there are other sources for the trends that cannot be quantified by any specific external parameters (for instance, there are thin clouds moving across the subsequent images), one can involve the method of Trend Filtering Algorithm (TFA; Kovács, Bakos & Noyes, 2005). This algorithm is based on the experience that there are stars with no intrinsic variability showing the same features in their light curves. TFA removes these trends by using a set of *template* stars (preferably none of them are variables) and searching for coefficients that can be used to perform a linear combination between the template light curves and then this best fit linear combination is subtracted from the original signal. Fig. 2.18 displays these two primary sources of the trends, in the case of some non-variable stars[22]. In the cases when analysis is performed on a photometric data set which does have only time series information about the magnitudes, the method of EPD cannot be applied while TFA still can be very effective (for a recent application, see e.g. Szulágyi, Kovács & Welch, 2009).

Of course, there are several other methods found in the literature that are intended to remove or at least, decrease the amplitude of unexpected systematic variations in the light curves. The concept of the SysRem method (Tamuz, Mazeh & Zucker, 2005) can be summarized shortly as an algorithm that searches decorrelation coefficients similar to the ones used in the EPD simultaneously to all of the light curves then repeats this procedure by assuming the external parameters themselves to be unknowns. This method of SysRem has been improved by Cameron et al. (2006) in order to have a more robust and reliable generic transit search algorithm. The ad-hoc template selection of the TFA has been replaced by a hierarchical clustering algorithm by Kim et al. (2008), assuming that stars showing similar trends are somehow localized. In the following, we are focusing on the EPD and TFA algorithms, since in the HATNet data reductions these algorithms play a key role.

---

[22]These stars are suspected not to be variables above the noise limits of the measurements. The data displayed here originate from the first follow-up transit measurements of the HAT-P-7(b) planetary system on 2007 November 2. See Chapter 3 for further details about the related data reductions.

2.10.1 Basic equations for the EPD and TFA

Let us assume having a photometric time series for a particular star and denote the instrumental magnitudes by m_i ($i = 1, \ldots, N$ where N is the total number of data points). The external parameters involved in the decorrelation are denoted by $p_i^{(k)}$ ($k = 1, \ldots, P$, where P is the number of the independent external parameters) while the magnitudes template stars are $m_i^{(t)} + \bar{m}^{(t)}$ ($t = 1, \ldots, T$, where T is the total number of template stars and $\bar{m}^{(t)}$ is the mean magnitude for the template star t). The method of EPD then minimizes the merit function

$$\chi^2_{\text{EPD}} = \sum_i w_i \left(m_i - m_0 - \sum_k E_k p_i^{(k)} \right)^2, \tag{2.84}$$

where E_k's are the appropriate EPD coefficients, m_0 is the mean brightness of the star and the weight of the given photometric point i is w_i, usually $w_i = \sigma_i^{-2}$ (σ_i is the individual photometric uncertainty for the measurement i). One of the most frequently used $\mathbf{p}_i$ parameter vector used in the EPD of HATNet light curves is $\mathbf{p}_i = \{x_i - \bar{x}, y_i - \bar{y}, S_i, D_i, K_i, 1/\cos(z_i), \tau_i\}$, where x_i and y_i are the centroid coordinates on the original frames, S_i, D_i and K_i are the stellar profile shape parameters defined in equation (2.32), z_i is the zenith distance (thus, $1/\cos(z_i)$ is the airmass) and τ_i is the hour angle. The $\bar{q}$ refers to the average of the quantity q. Although the EPD method yields a linear equation for the coefficients E_k, omitting the subtraction of the average centroid coordinates might significantly offset the value of m_0 from the real mean magnitude. Due to the linearity of the problem, this is not relevant unless one wants to rely on the value of m_0 in some sense[23] The function that is minimized by TFA is

$$\chi^2_{\text{TFA}} = \sum_i w_i \left(m_i - m_0 - \sum_t F_t m_i^{(t)} \right)^2, \tag{2.85}$$

where the appropriate coefficient for the template star t is F_t. The similarities between equation (2.84) and equation (2.85) are obvious. Indeed, one can perform the two algorithms simultaneously, by minimizing the joint function of

$$\chi^2_{\text{E+T}} = \sum_i w_i \left(m_i - m_0 - \sum_k E_k p_i^{(k)} - \sum_t F_t m_i^{(t)} \right)^2. \tag{2.86}$$

The de-trended light curve is then

$$m_i^{(\text{EPD})} = m_i - \sum_k E_k p_i^{(k)}, \tag{2.87}$$

---

[23]For instance, light curves from the same source might have different average magnitudes in the case of multi-station observations. The average magnitudes are then shifted to the same level prior to the joint analysis of this photometric data. Either m_0 or the median value of the light curve magnitudes can be used as an average value.

$$m_i^{(\text{TFA})} = m_i - \sum_t F_t m_i^{(t)} \quad \text{or} \tag{2.88}$$

$$m_i^{(\text{EPD+TFA})} = m_i - \sum_k E_k p_i^{(k)} - \sum_t F_t m_i^{(t)}, \tag{2.89}$$

for EPD, TFA and the joint trend filtering, respectively.

2.10.2   Reconstructive and simultaneous trend removals

Of course, we are not really interested in the de-trending of non-variable stars. Unless one wants to quantify the generic quality of a certain photometric pipeline, the importance of any trend removal algorithm are relevant only in the cases where the stars *have* intrinsic brightness variations. In the following, we suppose that the physical variations can be quantified by a small set of parameters $\{A_r\}$, namely the fiducial signal of a particular star can be written as

$$m_i^0 = m_0 + F(t_i, A_1, \ldots, A_R) \tag{2.90}$$

where F is some sort of model function.

In principle, one can manage variable stars by four considerations. First, even stars with physical brightness variations are treated as non-variable stars. This naive method is likely to distort the signal shape by treating the intrinsic changes in the brightness to be unexpected. In the cases where the periodicity of these intrinsic variations are close to the periodicity of the generic trends[24] or when the period is comparable or longer with the observation window, either EPD or TFA tend to kill the real signal itself. Second, one can involve the method of *signal reconstruction*, as it was implemented by Kovács, Bakos & Noyes (2005). In this method, the signal model parameters $\{A_r\}$ are derived using the noisy signal, and then the fit residuals undergo either the EPD or TFA. The model signal $F(t_i, \ldots)$ is added to the de-trended residuals, yielding a complete signal reconstruction. The steps can be repeated until convergence is reached. Third, one can involve the *simultaneous* derivation of the A_r model parameters and the E_k/F_t coefficients by minimizing the merit function

$$\chi^2 = \sum_i w_i \left[m_i - m_0 - F(t_i, \{A_r\}) - \sum_k E_k p_i^{(k)} \right]^2. \tag{2.91}$$

(This merit function shows the simultaneous trend removal for EPD. The TFA and the joint EPD+TFA can be applied similarly.) The fourth method derives the E_k and/or T_f coefficients on sections of the light curve where the star itself shows no real variations. This is a definitely useful method in the analysis of planetary transit light curves, since the star itself can be assumed to have constant brightness within noise limitations[25] and therefore

---

[24]For instance, trends with a period of a day are generally very strong.

[25]At least, in the most of the cases. A famous counter-example is the star CoRoT-Exo-2 of Alonso et al. (2008).

the light curve should show no variations before and after the transit. If these out-of-transit sections of the light curves are sufficiently long, the trend removal coefficients E_k and/or T_f can safely be obtained.

There are some considerations regarding to the $F(t_i, A_1, \ldots, A_R)$ function and its parameters $\{A_r\}$ that should be mentioned here. In principle, one can use a model function that is related to the *physics* of the variations. For instance, a light curve of a transiting extrasolar planet host star can be well modelled by 5 parameters[26]: period (P), epoch (E), depth of the transit (d), duration of the transit (τ_{14}) and the duration of the ingresses/egresses (τ_{12}) (see e.g. Carter et al., 2008, about how these parameters are related to the physical parameters of the system, such as normalized semimajor axis, planetary radius and orbital inclination). Although the respective model function, $F_{\text{transit}}(t_i, P, E, d, \tau_{14}, \tau_{12})$ is highly non-linear in its parameters, the simultaneous signal fit and trend removal of equation (2.91) can be performed, and the fit yields reliable results in general[27]. In the cases where we do not have any *a priori* knowledge of the source of the variations, but the signal can be assumed to be periodic, one can use a periodic model for F, that is, for instance, a linear combination of step functions. Although the number of free parameters (which must be involved in such a fit) are significantly larger, in the cases of HATNet light curves, the fit can be achieved properly. The signal reconstruction algorithm of Kovács, Bakos & Noyes (2005) use a step function (also known as "folded and binned light curve models") for this purposes. Like so, F can also be written as a Fourier series with finite terms. If the period and epoch are kept fixed, both assumptions for the function F (i.e. step function or Fourier expansion) yield a linear fit for both the model parameters and the EPD/TFA coefficients.

It should be mentioned here that the signal reconstruction mode and the simultaneous trend removal yields roughly the same results. However, a prominent counter-example is the case of HAT-P-11(b) (Bakos, Torres, Pál et al., 2009), where the reconstruction mode yielded an unexpectedly high impact parameter for the system. In this case, only the method of simultaneous EPD and TFA was able to reveal a refined set of light curve parameters that are expected to be more accurate on an absolute scale. Further discussion of this problem can be found in Bakos, Torres, Pál et al. (2009).

---

[26]Other parameters might be present if we do not have *a priori* assumptions for the limb darkening and/or the planetary orbit is non-circular and the signal-to-noise of the light curve is sufficiently large to see the asymmetry.

[27]Only if the transit instances inter/extrapolated from the initial guess for the epoch E and period P sufficiently cover the observed transits. Otherwise, all of the parametric derivatives of F will be zero and only methods based on systematic grid search (e.g. BLS) yield reliable results.

2.10.3 Efficiency of these methods

It is important to emphasize that both the EPD and TFA algorithms (independently from their native, reconstructive or simultaneous applications) reduce the effective degrees of freedom and therefore the light curve scatter *always* decreases. In order to determine whether the application of any of these algorithms is effective, one should compute the unbiased residuals of the fit after the derivation of the de-correlation coefficients. Alternatively, one can increase the scatter of a particular light curve by the factor $\sqrt{N/(N-P)}$ where N is the number of total data points in the light curve and P is the number of parameters involved in the EPD or TFA. We should keep in mind that both during the selection of the appropriate external parameters and during the template selection, the unbiased residuals must be checked carefully, otherwise the efficiency of these algorithms can easily be overrated.

2.11 Major concepts of the software package

Continuous monitoring of the sky yields enormous amount of data. In the HATNet project, 6 telescopes expose images with a cadence of 5.5 minutes. Each image is a $2\,\mathrm{k} \times 2\,\mathrm{k}$ (up to August 2007) or $4\,\mathrm{k} \times 4\,\mathrm{k}$ array of pixels, thus the amount of data gathered on each clean night is $\sim 80 - 120$ scientific frames for a single telescope, equivalent to $7 - 11$ or $30 - 45$ gigabytes of uncompressed calibrated images (assuming frames with the size of $2\,\mathrm{k} \times 2\,\mathrm{k}$ or $4\,\mathrm{k} \times 4\,\mathrm{k}$ pixels, respectively). In other words, if a single field is monitored for 2 months by two of the telescopes (see e.g. Bakos et al., 2007b, for a description of the actual observational principles), yielding ~ 5000 individual scientific frames. The amount of data associated to this certain field is $\sim 300 - 350$ gigabytes in a form of calibrated images (assuming $4\,\mathrm{k} \times 4\,\mathrm{k}$ images). If photometry is performed on these frames, the amount of associated information for 10 000 stars and for a single frame is ~ 3 megabytes of data, therefore one needs hundreds of gigabytes storage space just for the photometric results. All in all, the total amount of data that can be associated to the reduction of a single monitored field can be even be close to one terabyte, including all of the results of previously mentioned data types as well as other ones, for instance astrometrical information, subtracted images, or light curves with some sort of de-trending.

The components of the software package must be appropriate to manage such a huge amount of data. Thus, before going into the details of the practical implementation, two issues should be clarified. First, what kinds of data structures do appear during the reduction of the images? This is a rather important question since the programs not only have to access and manipulate these data but the resource limitations of the computers do also constrain the available solutions. Second, what are the existing software solutions which can efficiently be exploited? We are especially focusing on such operating systems and the related tools

Table 2.3: Comparison of various data storage schemes. In this list, "blobs" are used as an acronym for "binary large objects" (a collection of purely binary data in a single file). .

Pros	Cons
FITS[1] • Flexible Image Transport System. The most common format and standard for astronomical data storage, especially for images (either raw or calibrated) and spectra. • Extensible format, supports not only multidimensional numeric arrays but in addition, structured flat tables and ASCII tables can also be stored inside a single FITS file. • Metadata storage is also available in a form of keywords and their associated values. For instance, the location of observation, information about the observer, date and time of the observation, instrumental details (such as filters, exposure time, optical data for the telescope) are stored almost always in FITS files involving consensual keywords.	• Although some parts of the FITS files are stored in ASCII form (such as keywords and their values and textual tables), extracting data from FITS files requires special tools. Moreover, to access to just a smaller segment of the FITS data, one likely has to parse the whole file. For instance, if in a single file there are 100 stored tables and one needs data only from the last table, all of the other tables (at least their headers) have to be read and parsed, since there is no pre-defined location of the last table. • Inserting or removing some keywords to/from a FITS header likely results in an update of the whole file.
Binary (large) objects • Binary large object files (also known as "blobs") provide the fastest way for both accessing (reading) and writing data. • Various indexing algorithms are available to make the data access more efficient. Such algorithms can be optimized for any kind of data structure and access mode, including sequential access or two- or multidimensional hierarchy of data records.	• Such blobs are *not* human-readable, special programs are required for accessing, reading or modifying the data. Basic tools found in UNIX-like systems are not capable for generic manipulation of binary data. • Binary representation of integers and floating point numbers depends on the actually used computer/processor architecture. Unless special attention is given, such blobs cannot be copied from one computer to another if they are using different architectures. Involving an architecture independent storage format reduces some advantages of blobs (such as fast access).
Linear ASCII/text files • Human-readable format, easy to interpret. • Basic tools found in UNIX-like systems are capable to view or manipulate plain textual data. • All of the programming languages, including data processing environments and plotting tools support to read and parse numeric data from textual formats. • Modifications are easy to implement. Any kind of text editors or word processors are appropriate for manual manipulating of the data.	• Access to massive numeric data in stored in textual format can significantly be slower than access to the same amount of data that are stored in blobs. • Random or even non-sequential access of small chunks of data stored in a single text file requires the reading and parsing of the whole file. • The same type of data require $5-8$ times larger storage space than if these data were stored in blobs (depending on the actual data types and/or our needs for a well structured file).
Third-party applications: database servers, external storage systems • Easy to maintain. Database solutions support various methods for management, access control and such servers come with programming interfaces for many languages and environments. • The underlying database engines involve large number of algorithms for optimal data storage and allow efficient queries (using various indexing methods). The engines can be fine-tuned in order to optimize for our particular problem.	• The indexing and therefore the access to the data is optimized for one dimensional arrays of records (i.e. "flat" tables). Therefore storing images or other two- or multidimensional data structures (such as astronomical catalogs, long photometric time series of enormous amount of objects) cannot be implemented efficiently using classical database engines.

---

[1] The detailed documentation about the FITS file format is available from `http://fits.gsfc.nasa.gov/`.

that are supported by larger communities and have a free and portable implementation.

2.11.1 Data structures

At a first glance, data associated with image reduction can be classified into two major groups. The first group, that requires the most of the storage space is in the form of massive linear data, such as sequences of records, arrays of basic types or other multidimensional arrays. Astronomical images, processed images (such as registered or subtracted ones), instrumental photometric information, light curves, de-trended light curves, Fourier or other kind of spectra of the light curves belong to this group. All of these data are a set of records with the same structure. For instance,

- an image is a two dimensional array of integer or real numbers;

- the list of extracted sources, where each record contains information on the source's coordinates, brightness, shape parameters and possible catalogue identifiers;

- a light curve is a series of individual photometric measurements, where each measurement has a time, some sort of quality flag, magnitudes for various apertures and/or various photometric methods, uncertainty estimations; or

- instrumental photometry, where the records contain the same kind of information as the records of light curves, but one set of records is associated not to a particular object covering a long timebase, but to a single frame and numerous individual objects;

- additional catalogue information for each star, that can be useful in the interpretation of the photometric time series: such as brightness, color, spectral type, evolutionary state, parallax (if known), variability (if known).

These data types in the following are referred to as simply "data" in a general context.

The second major group of data types is the "metadata", that do not have linear structure like the data types discussed above, and represent definitely smaller amount of information. For instance,

- observational conditions for each image, such as date and time of the observation, location, instrument description, primary target object or field;

- astrometric solution, where the information itself is the transformation that maps a reference catalogue to the frame of the image;

- point-spread function for a single image;

- kernel solution, that describes the convolution function used in the process of image subtraction.

Table 2.4 summarizes the above mentioned various data types and their expected storage space requirements appearing in the photometric analysis.

Of course, both linear data and metadata that are created during the image reduction process should be stored in some format. There are various concepts for data formats available in modern computers and operating systems, so one can choose the most suitable format for each purpose. In astronomy, people commonly store and share data in FITS format. Many programs use human-readable (ASCII or text) files both for input and output. Some other programs store their information in binary format, where the contents of the files cannot even be viewed without a special program. And there are robust database systems, that hide the details of the actual storage and give a relatively lightweight interface to access or manipulate the data. Each type of the above mentioned data representations has its own advantages and disadvantages. In Table 2.3 these properties are summarized for these four major representation schemes. During the reduction of HATNet data, we have chosen a mixed form of data representation as follows. The images, including the raw, calibrated and processed ones are stored in FITS format. Moreover, we use three dimensional FITS images to store the spatial variations of the point-spread function. Other metadata, such as astrometrical solutions, kernel solutions, catalogue information are stored in text files. Instrumental photometric measurements and light curves are also stored in the form of text files. Temporary data (needed for intermediate steps of the reduction) are stored in binary form, since such data are not needed to be portable and an advantage of the binary format is the significantly smaller storage space requirement.

2.11.2 Operating environment

In order to both have a portable and robust set of tools, one has to build a software package on the top of widely standardized and documented environment. The most widespread and approved standard is the "Portable Operating System Interface" or POSIX[28], that intended to standardize almost all layers of the operating system, from the system-level application program interfaces (APIs, such as file manipulation or network access) up to the highest level of programs such as shell environments, related scripting languages and other basic utilities.

The actual development of the package FI/FIHAT was done under GNU[29]/Linux[30] sys-

---

[28]http://en.wikipedia.org/wiki/POSIX
[29]http://www.gnu.org/
[30]http://www.kernel.org/

Table 2.4: An overview of data files used to store information required by the image reduction process or created during the reduction. Each file type is referred by its extension.

File size ($\rightarrow$) # of files($\downarrow$)	$\mathcal{O}(1)$	$\propto N_{\mathrm{frame}}$	$\propto N_{\mathrm{object}}$
$\mathcal{O}(1)$	`*.config`: generic information about the whole reduction and observational conditions (name and coordinates of the target field, involved reduction algorithms and their fine-tune parameters)	`*.list`: list of frames to be processed during the reduction `*.stat`: basic statistics for each frame (both image statistics such as number of detected objects and information on the observational circumstances, e.g. zenith distance, airmass, elevation of the Moon, stellar profile FWHM)	`*.cat`: list of objects and some catalogue information that is used during the reduction `*.lcstat`: light curve statistics (also known as "magnitude-rms" statistics)
$\propto N_{\mathrm{frame}}$	`*.trans`: astrometric solution (the transformation that maps the reference catalogue to the coordinate system of the image) `*.kernel`: kernel solution (the convolution function used in the image subtraction process) `*-psf.fits`: best fit point-spread function for a given image	✕	`*.fits`: calibrated images[1] `*-sub.fits`: convolved and subtracted images[1] `*.stars`: list of detected sources and their properties (coordinates, shape parameters, brightness estimation) `*.phot`: instrumental photometric measurements
$\propto N_{\mathrm{object}}$	`*.xmmc`: best fit and Monte-Carlo distribution of the parameters of the light curve model function (if the object is turned out to be interesting) `*.info`: summary information of the planetary, orbital and stellar data for the actual object (if the object is indeed a planet-harboring star)	`*.lc`: light curves `*.epdlc`: de-trended light curves involving only the External Parameter Decorrelation algorithm `*.tfalc`: de-trended light curves involving the Trend Filter Algorithm	✕

[1] Strictly speaking, the size of these files does not depend on the number of objects that are extracted from the image and/or targets for further photometry. However, larger images tend to have greater number of sources of interest.

tems, that is one of the most frequently used POSIX compliant, UNIX-like[31] free operating system. The main code was written in ANSI C (featured with some GNU extensions) and intended to be compiled without any difficulties on various other UNIX systems such as SunOS/Sparc and Mac OSX. The compilation of the package does not require additional packages or libraries, only the GNU C Compiler (`gcc`[32]), its standard library (`glibc`[33]), the associated standard header files and some related development utilities. (Such as `make`[34] or the `ar`[35] object archived. In almost all of the systems these come with `gcc` as its dependencies) Therefore, all of the requirements of the package include only free and open source software (F/OSS).

In practice, to have a complete data reduction environment the users of the package might have to use additional text processing utilities such as an implementation of the AWK programming language (for instance, `gawk`[36], that is included in all of the free GNU/Linux systems) and basic text processing utilities (such as `paste`, `cat`, `sort`, `split`, included in the `textutils`/`coreutils`[37] GNU package). And finally, for visualization purposes, the SAOImage/DS9 utility[38] (Joye & Mandel, 2003) is highly recommended.

2.12   Implementation

In this subsection I summarize the standalone programs that are implemented as distinct binary executables. The programs can be divided into two well separated groups with respect to the main purposes. In the first group there are the programs that manipulate the (astronomical) images themselves, i.e. read an image, generate one or do a specific transformation on an image. In the second group, there are the programs that manipulate textual data, mostly numerical data presented in a tabulated form.

Generally, all of these programs are capable to the following.

- The codes give release and version information as well as the invocation can be logged on demand. The version information can be reported by a single call of the binary, moreover it is logged along with the invocation arguments in the form of special FITS keywords (if the main output of the actual code is a processed FITS image) and in the form of textual comments (if the main output of the code is text data). Preserving

---

[31]http://en.wikipedia.org/wiki/Unix-like
[32]http://gcc.gnu.org/
[33]http://www.gnu.org/software/libc/
[34]http://www.gnu.org/software/make/
[35]http://www.gnu.org/software/binutils/
[36]http://www.gnu.org/software/gawk/
[37]http://www.gnu.org/software/coreutils/
[38]http://hea-www.harvard.edu/RD/ds9/

Table 2.5: An overview of the standalone binary programs included in the package, displaying their main purposes and the types of input and output data.

Program	Main purpose	Type of input	Type of output
`fiarith`	Evaluates arithmetic expressions on images as operands.	A set of FITS images.	A single FITS image.
`ficalib`	Performs various calibration steps on the input images.	A set of raw FITS images.	A set of calibrated FITS image.
`ficombine`	Combines (most frequently averages) a set of images.	A set of FITS images.	A single FITS image.
`ficonv`	Obtains an optimal convolution transformation between two images or use an existing convolution transformation to convolve an image.	Two FITS images or a single image and a transformation.	A convolution transformation or a single image.
`fiheader`	Manipulates, i.e. reads, sets, alters or removes some FITS header keywords and/or their values.	A single FITS image (alternation) or more FITS images (if header contents are just read).	A FITS image with altered header or a series of keywords/values from the headers.
`fiign`	Performs low-level manipulations on masks associated to FITS images.	A single FITS image (with some optional mask).	A single FITS image (with an altered mask).
`fiinfo`	Gives some information about the FITS image in a human-readable form or creates image stamps in a conventional format.	A single FITS image.	Basic information or PNM images.
`fiphot`	Performs photometry on normal, convolved or subtracted images.	A single FITS image (with additional reference photometric information if the image is a subtracted one).	Instrumental photometric data.
`firandom`	Generates artificial object lists and/or artificial (astronomical) images.	List of sources to be drawn to the image or an arithmetic expression that describes how the list of sources is to be created.	List of sources and/or a single FITS image.
`fistar`	Detects and characterizes point-like sources from astronomical images.	A single FITS image.	List of detected sources and an optional PSF image (in FITS format).
`fitrans`	Performs generic geometric (spatial) transformations on the input image.	A single FITS image.	A single, transformed FITS image.
`fi[un]zip`	Compresses and decompresses primary FITS images.	A single uncompressed or compressed FITS image file.	A single compressed or uncompressed FITS image file.
`grcollect`	Performs data transposition on the input tabulated data or do some sort of statistics on the input data.	A set of files containing tabulated data.	A set of files containing the transposed tabulated data or a single file for the statistics, also in a tabulated form.
`grmatch`	Matches lines read from two input files of tabulated data, using various criteria (point matching, coordinate matching or identifier matching).	Two files containing tabulated data (that must be two point sets in the case of point or coordinate matching).	One file containing the matched lines and in the case of point matching, an additional file that describes the best fit geometric transformation between the two point sets.
`grselect`	Selects lines from tabulated data using various criteria.	A single file containing tabulated data.	The filtered rows from the input data.
`grtrans`	Transforms a single coordinate list or derives a best-fit transformation between two coordinate lists.	A single file containing a coordinate list and a file that describes the transformation or two files, each one is containing a coordinate list.	A file with the transformed coordinate list in tabulated from or a file that contains the best-fit transformation.
`lfit`	General purpose arithmetic evaluation, regression and data analysis tool.	Files containing data to be analyzed in a tabulated form.	Regression parameters or results of the arithmetic evaluation.

the version information along with the invocation arguments makes any kind of output easily reproducible.

- All of the codes are capable to read their data to be processed from the standard input and write the output data to the standard output. Since many of these programs manipulate relatively large amount of data, the number of unnecessary hard disk operations should be reduced as small as possible. Moreover, in many cases the output of one of the programs is the input of the another one. Pipes, available in all of the modern UNIX-like operating systems, are basically designed to perform such bindings between the output and input of two programs. Therefore, such a capability of redirecting the input/output data streams significantly reduce the overhead of background storage operations.

- The programs that deal with symbolic operations and functions, a general back-end library[39] is provided to make a user-friendly interface to specify arithmetic expressions. This kind of approach in software systems is barely used, since such a symbolic specification of arithmetic expressions does not provide a standalone language. However, it allows an easy and transparent way for arbitrary operations, and turned out to be very efficient in higher level data reduction scripts.

- The programs that manipulate FITS images are capable to handle files with multiple extensions. The FITS standard allows the user to store multiple individual images, as well as (ASCII or binary) tabulated data in a single file. The control software of some detectors produces images that are stored in this extended format, for example, such detectors where the charges from the CCD chip are read out in multiple directions (therefore the camera electronics utilizes more than one amplifier and A/D converter, thus yield different bias and noise levels). Other kind of detectors (which acquire individual images with a very short exposure time) might store the data in the three dimensional format called "data cube". The developed codes are also capable to handle such data, therefore it is possible to do reductions on images obtained by the Spitzer Space Telescope, that optionally uses such data structures for image storage.

The list of standalone binaries and their main purposes that come with the package are shown in Table 2.5.

2.12.1 Basic operations on FITS headers and keywords – `fiheader`

The main purpose of the `fiheader` utility is to read specific values from the headers of FITS files and/or alter them on demand.

---

[39]available from http://libpsn.sf.net

Although most of the information about the observational conditions is stored in the form of FITS keywords, image manipulation programs use only the necessary ones and most of the image processing parameters are passed as command line arguments (such keywords and data are, for example, the gain, the image centroid coordinates, astrometrical solutions). The main reasons why this kind of approach was chosen are the following.

- First, interpreting many of the standard keywords leads to false information about the image in the cases of wide-field or heavily distorted images. Such a parameter is the gain that can be highly inhomogeneous for images acquired by an optical system with non-negligible vignetting and the gain itself cannot be described by a single real number[40], rather a polynomial or some equivalent function. Similarly, the standard World Coordinate System information, describing the astrometrical solution of the image, has been designed for small field-of-view images, i.e. the number of coefficients are insufficiently few to properly constrain the astrometry of a distorted image.

- Second, altering the meanings of standard keywords leads to incompatibilities with existing software. For example, if the format of the keyword GAIN was changed to be a string of finite real numbers (describing a spatially varied gain), other programs would not be able to parse this redefined keyword.

Therefore, our conclusion was not altering the syntax of the existing keywords, but to define some new (wherever it was necessary). The fiheader utility enables the user to read any of the keywords, and allows higher level scripts to interpret the values read from the headers and pass their values to other programs in the form of command line arguments.

2.12.2 Basic arithmetic operations on images – fiarith

The program fiarith allows the user to perform simple operations on one or more astronomical images. Supposing all of the input images have the same size, the program allows the user to do *per pixel* arithmetic operations as well as manipulations depend on the pixel coordinates themselves.

The invocation syntax simply reflects the desired operations. For example the common way of calibrating image I, using bias (B), dark (D) and flat (F) images, which can be written as

$$C = \frac{I - B - D}{F/\|F\|}, \tag{2.92}$$

where C denotes the calibrated image (see also equation 2.1). Thus, the computation of the calibrated image C can be written as

---

[40]For which the *de facto* standard is the GAIN keyword.

```
fiarith "('input'-'Bias'-'Dark')/('Flat'/norm('Flat'))" -o calibrated
```

2.12.3 Basic information about images – `fiinfo`

The aim of the program `fiinfo` is twofold. First, this program is capable to gather some statistics and masking information of the image. These include

- general statistics, such as mean, median, minimum, maximum, standard deviation of the pixel values;

- statistics derived after rejecting the outlier pixels;

- estimations for the background level and its spatial variations;

- estimations for the background noise; and

- the number of masked pixels, detailing for all occurring mask types.

The most common usage of `fiinfo` in this statistical mode is to deselect those calibration frames that seem to be faulty (e.g. saturated sky flats, aborted images or so).

Second, the program is capable to convert astronomical images into widely used graphics file formats. Almost all of the scaling options available in the well known `DS9` program (see Joye & Mandel, 2003) have been implemented in `fiinfo`, moreover, the user can define arbitrary color palettes as well. In practice, `fiinfo` creates only images in PNM (portable anymap) format. Images stored in this format can then be converted to any of the widely used graphics file formats (such as JPEG, PNG), using existing software (e.g. `netpbm`, `convert`/ImageMagick). Figures in this thesis displaying stamps from real (or mock) astronomical images have also been created using this mode of the program.

2.12.4 Combination of images – `ficombine`

The main purpose of image combination is to create a single image with good signal-to-noise ratio from individual images with lower signal-to-noise ratio. The program `ficombine` is intended to perform averaging of individual images. In practice, the usage of this program is twofold. First, it is used to create the master calibration frames, as it is defined by equation (2.3), equation (2.5) and equation (2.7). Second, the reference frame required by the method of image subtraction is also created by averaging individual registered object frames (see also Sec. 2.12.11 about the details of image registration).

In the actual implementation, such combination is employed as a *per pixel* averaging, where the method of averaging and its fine tune parameters can be specified via command line arguments. The most frequently used "average values" are the mean and median values.

In many applications, rejection of outlier values are required, for instance, omitting pixels affected by cosmic ray events. The respective parameters for tuning the outlier rejection are also given as command line options. See Sec. 2.12.5 for an example about the usage of `ficombine`, demonstrating its usage in the calibration pipeline.

2.12.5 Calibration of images – `ficalib`

In principle, the program `ficalib` implements the evaluation of equation (2.1) in an efficient way. It is optimized for the assumption that all of the master calibration frames are the same for all of the input images. Because of this assumption, the calibration process is much more faster than if it was done independently on each image, using the program `fiarith`.

Moreover, the program `ficalib` automatically performs the overscan correction (if the user specifies overscan regions), and also trims the image to its designated size (by clipping these overscan areas). The output images inherit the masks from the master calibration images, as well as additional pixels might be masked from the input images if these were found to be saturated and/or bloomed. When a single chip camera uses multiple readout gates, amplifiers and A/D converters the images are stored in a so-called mosaic format (such as KeplerCam). The program `ficalib` is capable to combine these mosaic image regions into one single image.

In Fig. 2.19 a shell script is shown that demonstrates the usage of the programs `ficalib` and `ficombine` on a real-life application, namely how the images acquired by the FLWO KeplerCam[41] are completely calibrated.

2.12.6 Rejection and masking of nasty pixels – `fiign`

The aim of the program `fiign` is twofold. First, it is intended to perform low-level operations on masks associated to FITS images, such as removing some of the masks, converting between layers of the masks and merging or combining masks from separate files. Second, various methods exist with which the user can add additional masks based on the image itself. These additional masks can be used to mark saturated or blooming pixels, pixels with unexpectedly low and/or high values or extremely sharp structures, especially pixels that are resulted by cosmic ray events.

This program is a crucial piece in the calibration pipeline if it is implemented using purely the `fiarith` program. However, most of the functionality of `fiign` is also integrated in `ficalib` (see Sec. 2.12.5). Since `ficalib` much more efficiently implements the operations of the calibration than if these were implemented by individual calls of `fiarith`, `fiign` is used only occasionally in practice.

---

[41]See: http://www.sao.arizona.edu/FLWO/48/kep.primer.html

```sh
#!/bin/sh

# Names of the individual files storing the raw bias, flat and object frames are stored here:
BIASLIST=($SOURCE/[0-9]*.BIAS.fits)
FLATLIST=($SOURCE/[0-9]*.FLAT.fits)
OBJLIST=($SOURCE/[0-9]*.TARGET.fits)

# Calibrated images:  all the images are got an 'R' prefix and put in the appropriate directory:
R_BIASLIST=($(for f in ${BIASLIST[*]} ; do echo $MSTTMP/bias/R`basename $f` ; done))
R_FLATLIST=($(for f in ${FLATLIST[*]} ; do echo $MSTTMP/flat/R`basename $f` ; done))
R_OBJLIST=( $(for f in ${OBJLIST[*]} ; do echo $TARGET/R`basename $f` ; done))

# These below are KeplerCam specific data, defining the topology and geometry of the CCD itself.
# The camera has four readout registers and therefore four amplifiers and A/D converters as well.
MS_NAME=(IM1 IM2 IM3 IM4)
MS_OPAR=spline,order=3,iterations=2,sigma=3
MS_OVER=(area={2:0:7:1023,1034:0:1039:1023,2:0:7:1023,1034:0:1039:1023},${MS_OPAR})
MS_OFFS=(1024,1024 0,1024 1024,0 0,0)
MS_TRIM=image:[8:0:1031:1023]

M_ARGS="--mosaic size=[2048,2048]"
M_ARGS="$M_ARGS --mosaic [name=${MS_NAME[0]},$MS_TRIM,overscan=[${MS_OVER[0]}],offset=[${MS_OFFS[0]}]]"
M_ARGS="$M_ARGS --mosaic [name=${MS_NAME[1]},$MS_TRIM,overscan=[${MS_OVER[1]}],offset=[${MS_OFFS[1]}]]"
M_ARGS="$M_ARGS --mosaic [name=${MS_NAME[2]},$MS_TRIM,overscan=[${MS_OVER[2]}],offset=[${MS_OFFS[2]}]]"
M_ARGS="$M_ARGS --mosaic [name=${MS_NAME[3]},$MS_TRIM,overscan=[${MS_OVER[3]}],offset=[${MS_OFFS[3]}]]"

# The calibration of the individual bias frames, followed by their combination into a single master
# image:
ficalib   -i ${BIASLIST[*]} --saturation 50000 $M_ARGS -o ${R_BIASLIST[*]}
ficombine ${R_BIASLIST[*]} --mode median -o $MASTER/BIAS.fits

# The calibration of the individual flat frames, followed by their combination into a single master
# image:
ficalib -i ${FLATLIST[*]} --saturation 50000 $M_ARGS -o ${R_FLATLIST[*]} \
        --input-master-bias $MASTER/BIAS.fits --post-scale 20000
ficombine ${R_FLATLIST[*]} --mode median -o $MASTER/FLAT.fits

# The calibration of the object images:
ficalib -i ${OBJLIST[*]} --saturation 50000 $M_ARGS -o ${R_OBJLIST[*]} \
        --input-master-bias $MASTER/BIAS.fits --input-master-flat $MASTER/FLAT.fits
```

Figure 2.19: A shell script demonstrating the proper usage of the `ficalib` and `ficombine` programs on the example of the calibration of the KeplerCam mosaic images. The names for the files containing the input raw frames (both calibration frames and object frames) are stored in the arrays `$BIASLIST[*]`, `$FLATLIST[*]` and `$OBJLIST[*]`. The variable `$M_ARGS` contains all necessary information related to the specification of the mosaic topology and geometry as well as the overscan areas associated to each readout direction. The individual calibrated bias and flat frames are stored in the subdirectories of the `$MSTTMP` directory. These files are then combined to a single master bias and flat frame, that are used in the final step of the calibration, when the object frames themselves are calibrated. The final calibrated scientific images are stored in the directory `$TARGET`. Note that each flat frame is scaled after calibration to have a mean value of 20,000 ADU. In the case of dome flats, this scaling is not necessary, but in the case of sky flats, this steps corrects for the variations in the sky background level (during dusk or dawn).

2.12.7 Generation of artificial images – `firandom`

The main purpose of the program `firandom` is to create artificial images. These artificial images can be used either to create *model images* for real observations (for instance, to remove
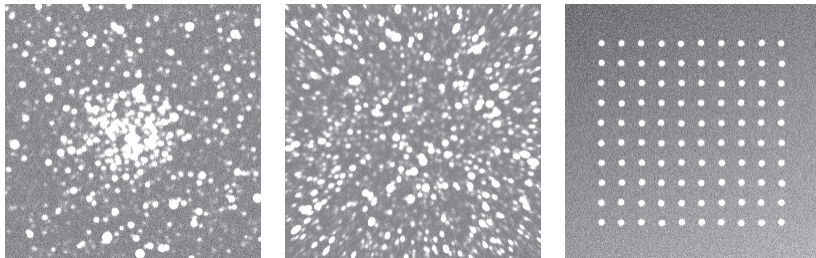
```sh
#!/bin/sh

firandom --size 256,256 \
        --list "f=3.2,500*[x=g(0,0.2),y=g(0,0.2),m=15-5*r(0,1)^2]" \
        --list "f=3.2,1400*[x=r(-1,1),y=r(-1,1),m=15+1.38*log(r(0,1))]" \
        --sky 100 --sky-noise 10 --integral --photon-noise --bitpix -32 --output globular.fits

firandom --size 256,256 \
        --list "5000*[x=r(-1,1),y=r(-1,1),s=1.3,d=0.3*(x*x-y*y),k=0.6*x*y,m=15+1.38*log(r(0,1))]" \
        --sky 100 --sky-noise 10 --integral --photon-noise --bitpix -32 --output coma.fits

firandom --size 256,256 \
        --list "f=3.0,100*[X=36+20*div(n,10)+r(0,1),Y=36+20*mod(n,10)+r(0,1),m=10]" \
        --sky "100+x*10-y*20" --sky-noise 10 --integral --photon-noise --bitpix -32 --output
grid.fits

for base in globular coma grid ; do
        fiinfo ${base}.fits --pgm linear,zscale --output-pgm - | pnmtoeps -g -4 -d -o ${base}.eps
done
```

Figure 2.20: Three mock images generated using the program `firandom`. The first image (`globular.fits`) on the left shows a "globular cluster" with some field stars as well. For simplicity, the distribution of the cluster stars are Gaussian and the magnitude distribution is quadratic while the field stars distribute uniformly and their magnitudes is derived from assuming uniformly distributed stars of constant brightness. The second image (`coma.fits`) simulates nearly similar effect on the stellar profiles what comatic aberration would cause. The shape parameters δ and κ (referred as `d` and `k` in the command line argument of the program, see also Sec. 2.4.2) are specific functions of the spatial coordinates. The magnitude distribution of the stars is the same as for the field stars in the previous image. The third image (`grid.fits`) shows a set of stars positioned on a grid. The background of this image is not constant. The shell script below is used to create these FITS files. The body of the last iterator loop in the script converts the FITS files into PGM format, using the `fiinfo` utility (see Sec. 2.12.3) and the well-known `zscale` intensity scaling algorithm (see DS9, Joye & Mandel, 2003). The images yielded by `fiinfo` are instantly converted to EPS (encapsulated Postscript) files, that is the preferred format for many typesetting systems, such as LaTeX.

fitted stellar PSFS) or *mock images* that are intended to simulate some of the influence related to one or more observational artifacts and realistic effects. In principle, `firandom` creates an image with a given background level on which sources are drawn. Additionally, `firandom` is capable to add noise to the images, simulating both the effect of readout and background noise as well as photon noise. In the case of mock images, `firandom` is also capable to generate the object list itself. The stellar profile models that are supported by

`firandom` and therefore available for artificial images are the same set of functions described in Sec.2.4.2. Moreover, `firandom` is capable to draw stellar profiles derived from PSFs (by the program `fistar`, see also Sec. 2.12.8).

The program features symbolic input processing, i.e. the variations in the background level, the spatial distribution of the object centroids (in the case of mock images), the profile shape parameters, fluxes for individual objects and the noise level can be specified not only as a tabulated dataset but in the form of arithmetic expressions. In these expressions one can involve various built-in arithmetic operators and functions, including random number generators. Of course, the generated mock coordinate lists can also be saved in tabulated form. The mock images used during the generation of Fig. 2.1, Fig. 2.2 or Fig. 2.5 have been created by `firandom`.

In Fig. 2.20, some examples are shown that demonstrate the usage of the program `firandom`.

2.12.8   Detection of stars or point-like sources – `fistar`

The star detection and stellar profile modelling algorithms described in Sec. 2.4 are implemented in the program `fistar`. The main purpose of this program is therefore to search for and characterize point-like sources. Additionally, the program is capable to derive the point-spread function of the image, and spatial variations of the PSF can also be fitted up to arbitrary polynomial order.

The list of detected sources, their centroid coordinates, shape parameters (including FWHM) and flux estimations are written to a previously defined output file. This file can have arbitrary format, depending on our needs. The best fit PSF is saved in FITS format. If the PSF is supposed to be constant throughout the image, the FITS image is a normal two-dimensional image. Otherwise, the PSF data and the associated polynomial coefficients are stored in "data cube" format, and the size of the z (`NAXIS3`) axis is $(N_{\mathrm{PSF}} + 1)(N_{\mathrm{PSF}} + 2)/2$, where N_{PSF} is the polynomial order used for fitting the spatial variations.

2.12.9   Basic coordinate list manipulations – `grtrans`

The main purpose of the program `grtrans` is to perform coordinate list transformations, mostly related to stellar profile centroid coordinates and astrometrical transformations. Since this program is used exhaustively with the program `grmatch`, examples and further discussion of this program can be found in the next section, Sec. 2.12.10.
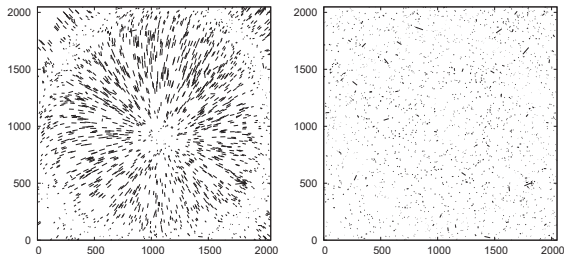
Figure 2.21: Vector plots of the difference between the transformed reference and the input star coordinates for a typical HAT field. The left panel shows the difference for second-order, the right panel for fourth-order polynomial fits.
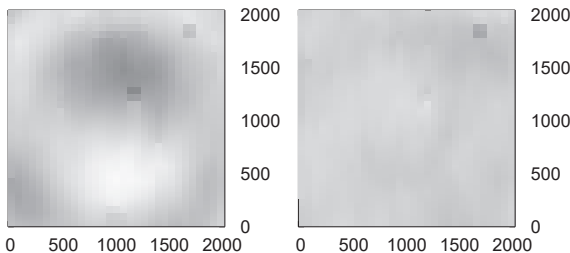
Figure 2.22: The difference between the Y coordinates of the transformed reference and the input star coordinates for a typical HAT field. The left panel shows the difference for fourth-order, the right panel for sixth-order polynomial fits.

2.12.10   Matching lists or catalogues – `grmatch`

The main purpose of the `grmatch` code is to implement the point matching algorithm that is the key point in the derivation of the astrometric solution and source identification. See Section 2.5 about more details on the algorithm itself. We note here that although the program `grmatch` is sufficient for point matching and source identification purposes, but one needs other codes to interpret or use the outcome of this program. For instance, tabulated list of coordinates can be transformed from one reference frame to another, using the program `grtrans` while the program `fitrans` is capable to apply these transformations (yielded by `grmatch`) on FITS images, in order to, for instance, register images to the same reference frame.

Typical applications

As it was discussed before, the programs `grmatch` and `grtrans` are involved in the photometry pipeline, following the star detection. If the accuracy of the coordinates in the reference catalogue is sufficient to yield a consistent plate solution, one can obtain the photometric

```
for base in ${LIST_OF_FRAMES[*]} ; do

        grmatch --reference $CATALOG --col-ref $COL_X,$COL_Y --col-ref-ordering -$COL_MAG \
                --input $AST/$base.stars --col-inp 2,3 --col-inp-ordering +8 \
                --weight reference,column=$COL_MAG,magnitude,power=2 \
                --order $AST_ORDER --max-distance $MAX_DISTANCE \
                --output-transformation $AST/$base.trans --output $AST/$base.match || break

        grtrans $CATALOG \
                --col-xy $COL_X,$COL_Y --input-transformation $AST/$base.trans \
                --col-out $COL_X,$COL_Y --output - | \
        grmatch --reference - --col-ref $COL_X,$COL_Y --input $AST/$base.stars --col-inp 2,3 \
                --match-coords --max-distance $MAX_MATCHDST --output - | \
        grtrans --col-xy $COL_X,$COL_Y --input-transformation $AST/$base.trans --reverse \
                --col-out $COL_X,$COL_Y --output $AST/$base.match

done
```

Figure 2.23: A typical application for the `grmatch` – `grtrans` programs, for the cases where a few of the stars have high proper motion thus have significant offsets from the catalogue positions. For each frame (named `$base`), the input catalogue (`$CATALOG`) is matched with the respective list of extracted stars (found in the `$AST/$base.stars` file), keeping a relatively large maximum distance between the nominal and detected stellar positions (`$MAX_DISTANCE`, e.g. $4-6$ pixels, derived from the expected magnitude of the proper motions from the catalogue epoch and the approximate plate scale). This first initial match identifies all of the sources (including the ones with large proper motion), stored in `$AST/$base.match` file in the form of matched detected source and catalogue entries. However, the astrometric transformation (stored in `$AST/$base.trans`) is systematically affected by these high proper motion stars. In order to get rid of this effect, the match is performed again by excluding the stars with higher residual distance (by setting `$MAX_MACHDIST` to e.g. $1-2$ pixels). The procedure is then repeated for all frames (elements of the `$LIST_OF_FRAMES[]` array) in the similar manner.

centroids by simply invoking these programs. A more sophisticated example for these program is shown in Fig. 2.23. In this example these programs are invoked twice in order to both derive a proper astrometric solution[42] and properly identify the stars with larger proper motions[43]. Such iterative invocation scheme is used frequently in case of the reduction of follow-up photometry data (see Chapter 4 and Sec. 4.1 for some other practical details). The simple direct application of `grmatch` and `grtrans` as a part of a complete photometric pipeline is displayed in Fig. 2.29.

2.12.11 Transforming and registering images – `fitrans`

As it was discussed earlier (Sec. 2.8), the image convolution and subtraction process requires the images to be in the same spatial reference system. The details of this registration process have been explained already in Sec. 2.6. The purpose of the program `fitrans` is to implement these various image interpolation methods.

---

[42]By taking into account only the stars with negligible proper motion.

[43]That would otherwise significantly distort the astrometric solution.

In principle, `fitrans` reads an image and a transformation file, performs the spatial transformation and writes the output image to a separate file. Image data are read from FITS files while the transformation files are presumably derived from the appropriate astrometric solutions. The output of the `grmatch` and `grtrans` programs can be directly passed to `fitrans`. Of course, `fitrans` takes into account the masks associated to the given image as well as derive the appropriate mask for the output file. Pixels which cannot be mapped from the original image have always a value of zero and these are marked as *outer pixels* (see also Sec. 2.3.2).

In the HATNet data reduction, this spatial transformation requires significant amount of CPU time since the exact integration on biquadratic interpolation surfaces is a computationally expensive process (Sec. 2.6.3). However, distinct image transformations can be performed independenlty (i.e. a given transformation does not have any influence on another transformations), thus the complete registration process can easily be performed in parallel.

2.12.12 Convolution and image subtraction – `ficonv`

This member of the FI/FIHAT package is intended to implement the tasks related to the kernel fit, image convolution and subtraction. In principle, `ficonv` has two basic modes. First, assuming an existing kernel solution, it evaluates equation (2.71) on an image and writes the convolved result to a separate image file. Second, assuming a base set of kernel functions (equation 2.73) and some model for the background variations (equation 2.75) it derives the best fit kernel solution for equation (2.71), described by the coefficients $C_{ik\ell}$ and $B_{k\ell}$, respectively. Since this fit yields a linear equation for these coefficients, the method of classic linear least squares minimization can be efficiently applied. However, the least squares matrix can have a relatively large dimension in the cases where the kernel basis is also large and/or higher order spatial variations are allowed. In the fit mode, the program yields the kernel solution, and optionally the convolved ($C = B + R \star K$) and the subtracted residual image ($S = I - C$) can also be saved into separate files without additional invocations of `ficonv` and/or `fiarith`.

The program `ficonv` also implements the fit for cross-convolution kernels (equation 2.79). In this case, the two kernel solutions are saved to two distinct files. Subsequent invocations of `ficonv` and/or `fiarith` can then be used to analyze various kinds of outputs.

In Sec. 2.9 we were discussing the relevance of the kernel solution in the case when the photometry is performed on the residual (subtracted) images. The best fit kernel solution obtained by `ficonv` has to be directly passed to the program `fiphot` (Sec. 2.12.13) in order to properly take into account the convolution information during the photometry (equation 2.83).

```
SELF=$0; base="$1"
if [ -n "$base" ] ; then
        fitrans ${FITS}/$base.fits \
                --input-transformation ${AST}/$base.trans --reverse -k -o ${REG}/$base-trans.fits
else
        pexec -f BASE.list -e base -o - -u - -c -- "$SELF \$base"
fi
```

```
SELF=$0; base="$1"
if [ -n "$base" ] ; then
        KERNEL="i/4;b/4;d=3/4"
        ficonv  --reference ./photref.fits \
                --input ${REG}/$base-trans.fits --input-stamps ./photref.reg --kernel "$KERNEL" \
                --output-kernel-list ${AST}/$base.kernel --output-subtracted ${REG}/$base-sub.fits
else
        pexec -f BASE.list -e base -o - -u - -c -- "$SELF \$base"
fi
```

Figure 2.24: Two shell scripts demonstrating the invocation syntax of the `fitrans` and `ficonv`. Since the computation of the transformed and convolved images require significant amount of CPU time, the utility `pexec` (`http://shellpexec.sf.net`) is used to run the jobs in parallel on multiple CPUs.

2.12.13   Photometry − `fiphot`

The program `fiphot` is the main code in the FI/FIHAT package that performs the raw and instrumental photometry. In the current implementation, we were focusing on the aperture photometry, performed on normal and subtracted images. Basically, `fiphot` reads an astronomical image (FITS file) and a centroid list file, where the latter should contain not only the centroid coordinates but the individual object identifiers as well[44].

In case of image subtraction-based photometry, `fiphot` requires also the kernel solution (derived by `ficonv`). Otherwise, if this information is omitted, the results of the photometry are not reliable and consistent. See also Sec. 2.9 for further details about this issue.

In Fig. 2.29, a complete shell script is displayed, as an example of various FI/FIHAT programs related to the photometry process.

Currently, PSF photometry is not implemented directly in the program `fiphot`. However, the program `fistar` (Sec. 2.12.8) is capable to do PSF fitting on the detected centroids, although its output is not compatible with that of `fiphot`. Alternatively, `lfit` (see Sec. 2.12.16) can be used to perform profile fitting, if the pixel intensities are converted to ASCII tables in advance[45], however, it is not computationally efficient.

---

[44]If the proper object identification is omitted, `fiphot` assigns some arbitrary (but indeed unique) identifiers to the centroids, however, in practice it is almost useless.

[45]The program `fiinfo` is capable to produce such tables with three columns: a list of x and y coordinates followed by the respective pixel intensitie.

```
# ${PHOT}/IMG-1.phot:                          # ${LC}/STAR-01.lc:
IMG-1 STAR-01 6.8765 0.0012 C                  IMG-1 STAR-01 6.8765 0.0012 C
IMG-1 STAR-02 7.1245 0.0019 G                  IMG-2 STAR-01 6.8778 0.0012 C
IMG-1 STAR-03 7.5645 0.0022 G                  IMG-3 STAR-01 6.8753 0.0012 G
IMG-1 STAR-04 8.3381 0.0028 G

                                               # ${LC}/STAR-02.phot:
                                               IMG-1 STAR-02 7.1245 0.0019 G
# ${PHOT}/IMG-2.phot:                          IMG-2 STAR-02 7.1245 0.0020 G
IMG-2 STAR-01 6.8778 0.0012 C                  IMG-3 STAR-02 7.1269 0.0019 G
IMG-2 STAR-02 7.1245 0.0020 G
IMG-2 STAR-03 7.5657 0.0023 G        ⇒
IMG-2 STAR-04 8.3399 0.0029 G                  # ${LC}/STAR-03.lc:
                                               IMG-1 STAR-03 7.5645 0.0022 G
                                               IMG-2 STAR-03 7.5657 0.0023 G
                                               IMG-3 STAR-03 7.5652 0.0023 G
# ${PHOT}/IMG-3.phot:
IMG-3 STAR-01 6.8753 0.0012 G                  # ${LC}/STAR-04.lc:
IMG-3 STAR-02 7.1269 0.0019 G                  IMG-1 STAR-04 8.3381 0.0028 G
IMG-3 STAR-03 7.5652 0.0023 G                  IMG-2 STAR-04 8.3399 0.0029 G
IMG-3 STAR-04 8.3377 0.0029 G                  IMG-3 STAR-04 8.3377 0.0029 G
```

```
grcollect ${PHOT}/IMG-*.phot --col-base 2 --prefix ${LC}/ --extension lc --max-memory 256m

cat ${PHOT}/IMG-*.phot | grcollect - --col-base 2 --prefix ${LC}/ --extension lc --max-memory 256m
```

Figure 2.25: The schematics of the data transposition. Records for individual measurements are written initially to photometry files (having an extension of `*.phot`, for instance). These records contain the source identifiers. During the transposition, photometry files are converted to light curves. In principle, these light curves contain the same records but sorted into distinct files by the object names, not the frame identifiers. The command lines on the lower panel show some examples how this data transposition can be employed involving the program `grcollect`.

2.12.14 Transposition of tabulated data – `grcollect`

Raw and instrumental photometric data obtained for each frame are stored in separate files by default as it was discussed earlier (see Sec. 2.7, Sec. 2.9 and Sec. 2.12.13). We refer to these files as *photometric files*. In order to analyze the per-object outcome of our data reductions, one has to have the data in the form of *light curve files*. Therefore, the step of photometry (including the magnitude transformation) is followed immediately by the step of *transposition*. See Fig. 2.25 about how this step looks like in a simple case of 3 photometric files and 4 objects.

The main purpose of the program `grcollect` is to perform this transposition on the photometric data in order to have the measurements being stored in the form of light curves and therefore to be adequate for further per-object analysis (such as light curve modelling). The invocation syntax of `grcollect` is also shown in Fig. 2.25. Basically, small amount of information is needed for the transposition process: the name of the input files, the index of the column in which the object identifiers are stored and the optional prefixes and/or suffixes for the individual light curve file names. The maximum memory that the program is allowed to use is also specified in the command line argument. In fact, `grcollect` does not

Figure 2.26: Storage schemes for photometric data. Supposing a series of frames, on which nearly the same set of stars have individual photometric measurements, the figure shows how these data can be arranged for practical usages. The target stars (their identifiers) are arranged along the abscissa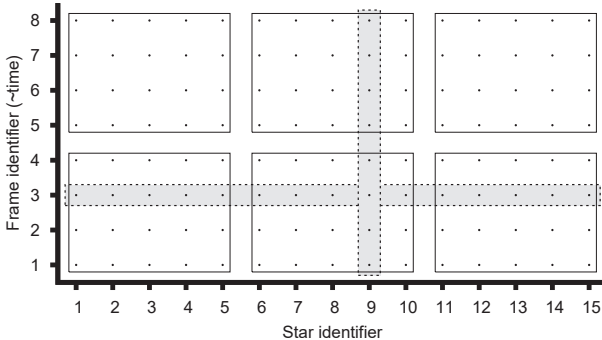 while the ordinate shows the frame identifiers to which individual measurements (symbolized by dots) belong. Raw and instrumental photometric data are therefore represented here as rows (see the marked horizontal stripe for frame #3, for instance) while the columns refer to light curves. In practice, native ways of transposition are extremely ineffective if the total amount of data does not fit into the memory. The transposition can be speeded up by using an intermediate stage of data storage, so-called macroblocks. In the figure, each macroblock is marked by an enclosing rectangle. See text for further details.

need the original data to be stored in separate files. The second example on Fig. 2.25 shows an alternate way of performing the transposition, namely when the whole data is read from the standard input (and the preceding command of `cat` dumps all the data to the standard output, these two commands are connected by a single uni-directional pipe).

The actual implementation of the transposition inside `grcollect` is very simple: it reads the data from the individual files (or from the standard input) until the data fit in the available memory. If this temporary memory is full of records, this array is sorted by the object identifier and the sorted records are written/concatenated to distinct files. The output files are named based on the appropriate object identifiers. This procedure is repeated until there are available data. Although this method creates the light curve files, it means that neither the whole process nor the access to these light curve files is effective. In case of HATNet, when we have thousands of frames in a single reduction and there are several tens or hundreds of thousands individual stars that are intended to have photometric measurements and each record is quite long[46], the total amount of data is in the order of hundreds of gigabytes. For even modern present-day computers, such a large amount of data does not fit in the memory. Therefore, referring to the simple process discussed above, light curve files are not written

---

[46]A record for a single photometric measurement is several hundreds of bytes long since it contains information for multiple apertures (including flux error estimations and quality flags) as well as there are additional fields for the stellar profile parameters and other observational quantities used in further trend filtering.

to the disk at once but in smaller chunks. These chunks are located on different cylinders of the disk: files are therefore extremely fragmented. Both the creation and the access of these fragmented files are extremely inefficient, since fragmented files require additional highly time-consuming disk operations such as random seeks between cylinders. In practice, even on modern computers (being used by the project), the whole process requires a day or so to be completed, although the sequential access to some hundreds of gigabytes of data would require only an hour or a few hours (with a plausible I/O bandwidth of $\sim 50\,\mathrm{MB/sec}$). In order to overcome this problem, one can either use an external database engine that features optimizations for such two-dimensional queries or tweak the above transposition algorithm to avoid unexpected and/or expensive disk operations. Now we briefly summarize an approach how the transposition can be made more effective if we consider some assumptions for the data structure. The program `grcollect` is capable to do transpositions even if some of the keys (stellar identifiers) are missing or if there are more than one occurrences for a single key in a given file. Let us assume that 1) in each input file every stellar identifier is unique and 2) the number of missing keys is negligible compared to the total number of photometric data records[47]. Assuming a total of N_F frames and $N_\star$ unique stellar identifiers (in the whole photometric data), the total number of records is $N_R \lesssim N_F N_\star$. The total memory capacity of the computer is able to store M records simultaneously. Let us denote the average disk seek time by τ and the sequential access speed by ω (in the units of records per second). The transposition can then be performed effectively in two stages. In the first stage the photometry files are converted to individual files, so-called macroblocks, where each of them is capable to store $(M/N_F) \times (M/N_\star)$ records, each macroblock represent a continuous rectangle in the stellar identifier – frame space (see Fig. 2.26). In the second stage, macroblock files are converted into light curves. Due to the size of the macroblock, $MN_F/N_\star$ photometric files can be read up sequentially and stored in the memory at the same time. If the relation

$$1 \ll \frac{M^2}{\tau N_f N_\star \omega} \tag{2.93}$$

is true for the actual values of M, N_f, $N_\star$, ω and τ, the macroblocks can be accessed randomly after the first stage (independently from the order in which they have been written to the disk), without too much dead time due to the random seeks. Therefore, at the second stage when macroblocks are read in the appropriate order of the stellar identifiers, $MN_\star/N_F$ light curves can be flushed simultaneously without any additional disk operations beyond sequential writing.

In the case of the computers used in HATNet data reduction, $M \approx 10^7$, $N_f \approx 10^4$, $N_\star \approx 10^5$, $\omega \approx 10^5$ records/sec and $\tau \approx 10^{-2}$ sec, the right-hand side of equation (2.93) is

---

[47]Each record represents a single photometric measurement for a single instant, including all additional relevant data (such as the parameters involved in the EPD analysis, see earlier)

Table 2.6: Algorithms supported by `lfit` and their respective requirements for the model function. The first column refers to the internal and command line identifier of the algorithms. The second column shows whether the method requires the parametric derivatives of the model functions in an analytic form or not. The third column indicates whether in the cases when the method requires parametric derivatives, should the model function be linear in all of the parameters.

Code	derivatives	linearity	Method or algorithm
L/CLLS	yes	yes	Classic linear least squares method
N/NLLM	yes	no	(Nonlinear) Levenberg-Marquardt algorithm
U/LMND	no	no	Levenberg-Marquardt algorithm employing numeric parametric derivatives
M/MCMC	no	no	Classic Markov Chain Monte-Carlo algorithm[1]
X/XMMC	yes	no	Extended Markov Chain Monte-Carlo[2]
K/MCHI	no	no	Mapping the values χ^2 on a grid (a.k.a. "brute force" minimization)
D/DHSX	optional[3]	no	Downhill simplex
E/EMCE	optional[4]	optional[4]	Uncertainties estimated by refitting to synthetic data sets
A/FIMA	yes	no	Fisher Information Matrix Analysis

[1] The implemented transition function is based on the Metropolitan-Hastings algorithm and the optional Gibbs sampler. The transition amplitudes must be specified initially. Iterative MCMC can be implemented by subsequent calls of `lfit`, involving the previous inverse statistical variances for each parameters as the transition amplitudes for the next chain.

[2] The also program reports the summary related to the sanity checks (such as correlation lengths, Fisher covariance, statistical covariance, transition probabilities and the best fit value obtained by an alternate /usually the downhill simplex/ minimization).

[3] The downhill simplex algorithm may use the parametric derivatives to estimate the Fisher/covariance matrix for the initial conditions in order to define the control points of the initial simplex. Otherwise, if the parametric derivatives do not exist, the user should specify the "size" of the initial simplex somehow during the invocation of `lfit`.

[4] Some of the other methods (esp. CLLS, NLLM, DHSX, in practice) can be used during the minimization process of the orignal data and the individual synthetic data sets.

going to be $\approx 10^2$, so the discussed way of two-stage transposition is very efficient. Indeed, the whole operation can be completed within $3 - 5$ hours, instead of a day or few days that is needed by the normal one-stage transposition. Moreover, due to the lack of random seeks, the computer itself remains responsible for the user interactions. In the case of one-stage transposition, the extraordinary amount of random seeks inhibit almost any interactive usage.

2.12.15 Archiving – `fizip` and `fiunzip`

Due to the large disk space required to store the raw, calibrated and the derived (registered and/or subtracted) frames, it is essential to compress and archive the image files that are barely used. The purpose of the `fizip` and `fiunzip` programs is to compress and decompress primary FITS data, by keeping the changes in the primary FITS header to be minimal. The compressed data is stored in a one-dimensional 8 bit (`BITPIX=8`, `NAXIS=1`) array, therefore these keywords does not reflect the original image dimension or data type.

All of the other keywords are untouched. Some auxiliary information on the compression

is stored in the keywords starting with "FIZIP", the contents of these keywords depend on the involved compression method. `fizip` rejects compressing FITS file where such keywords exist in the primary header.

In practice, `fizip` and `fiunzip` refer to the same program (namely, `fiunzip` is a symbolic link to `fizip`) since the algorithms involved in the compression and decompression refer to the same codebase or external library. `fizip` and `fiunzip` support well known compression algorithms, such as the GNU zip ("gzip") and the block-sorting file compressor (also known as "bzip2") algorithm.

These compression algorithms are lossless. However, `fizip` supports rounding the input pixel values to the nearest integer or to the nearest fraction of some power of 2. Since the common representation of floating-point real numbers yields many zero bits if the number itself is an integer or a multiple of power of 2 (including fractional multiples), the compression is more effective if this kind of rounding is done before the compression. This "fractional rounding" yields data loss. However, if the difference between the original and the rounded values are comparable or less than the readout noise of the detector, such compression does not affect the quality of the further processing (e.g. photometry).

2.12.16 Generic arithmetic evaluation, regression and data analysis − `lfit`

Modeling of data is a prominent step in the analysis and interpretation of astronomical observations. In this section, a standalone command line driven tool, named `lfit` is introduced, designed for both interactive and batch processed regression analysis as well as generic arithmetic evaluation.

This tool is built on the top of the `libpsn` library[48], a collection of functions managing symbolic arithmetic expressions. This library provides both the back-end for function evaluation as well as analytical calculations of partial derivatives. Partial derivatives are required by most of the regression methods (e.g. linear and non-linear least squares fitting) and uncertainty estimations (e.g. Fisher analysis). The program features many built-in functions related to special astrophysical problems. Moreover, it allows the end-user to extend the capabilities during run-time using dynamically loaded libraries.

In general, `lfit` is used extensively in the data reduction steps of the HATNet project. The program acts both in the main "discovery" pipeline and it is involved in the characterization of follow-up data, including photometric and radial velocity measurements. Currently, `lfit` implements executively the EPD algorithm (including the normal, the reconstructive and the simultaneous modes) as well as the simultaneous TFA algorithm (see e.g. Bakos,

---

[48]http://libpsn.sf.net, developed by the author

Table 2.7: Basic functions found in the built-in astronomical extension library. These functions cover the fields of simple radial velocity analysis, some aspects of light curve modelling and data reduction. These functions are a kind of "common denominators", i.e. they do not provide a direct possibility for applications but complex functions can be built on the top of them for any particular usage. All of the functions below with the exception of hjd() and bjd() have partial derivatives that can be evaluated analytically by lfit.

Function	Description
hjd(JD, α, δ)	Function that calculates the heliocentric Julian date from the Julian day J and the celestial coordinates α (right ascension) and δ (declination).
bjd(JD, α, δ)	Function that calculates the barycentric Julian date from the Julian day J and the celestial coordinates α (right ascension) and δ (declination).
ellipticK(k)	Complete elliptic integral of the first kind.
ellipticE(k)	Complete elliptic integral of the second kind.
ellipticPi(k, n)	Complete elliptic integral of the third kind.
eoq(λ, k, h)	Eccentric offset function, 'q' component. The arguments are the mean longitude λ, in radians and the Lagrangian orbital elements $k = e \cos \varpi$, $h = e \sin \varpi$.
eop(λ, k, h)	Eccentric offset function, 'p' component.
ntiu(p, z)	Normalized occultation flux decrease. This function calculates the flux decrease during the eclipse of two spheres when one of the spheres has uniform flux distribution and the other one by which the former is eclipsed is totally dark. The bright source is assumed to have a unity radius while the occulting disk has a radius of p. The distance between the centers of the two disks is z.
ntiq(p, z, γ_1, γ_2)	Normalized occultation flux decrease when eclipsed sphere has a non-uniform flux distribution modelled by quadratic limb darkening law. The limb darkening is characterized by γ_1 and γ_2.

Torres, Pál et al., 2009).

User interface and built-in regression methods

Due to the high modularization and freedom in its user interface, the program lfit allows the user to compare the results of different regression analysis techniques. The program features 9 built-in algorithms at the moment, including the classic linear least squares minimization (Press et al., 1992), the non-linear methods (Levenberg-Marquard, downhill simplex, see also Press et al., 1992), various methods providing an *a posteriori* distribution for the adjusted parameters, such as Markov Chain Monte-Carlo (Ford, 2004), or the method of refitting to synthetic data sets (Press et al., 1992). The program is also capable to derive the covariance or correlation matrix of the parameters involving the Fisher information analysis (Finn, 1992). The comprehensive list of the supported algorithms can be found in Table 2.6.

The basic concepts of lfit is shown in Fig. 2.27 in a form of a complete example for linear regression.

Built-in functions related to astronomical data analysis

The program `lfit` provides various built-in functions related to astronomical data analysis, especially ones that are required by exoplanetary research. All of these functions are some sort of "base functions", with a few parameters from which one can easily form more useful ones using these capabilities of `lfit`. Good examples are the eccentric offset functions $p(\lambda, k, h)$ and $q(\lambda, k, h)$ (Sec. 4.3), that have only three parameters but the functions related to the radial velocity analysis can easily be defined using these two functions. The full list of these special functions can be found in Table 2.7. The actual implementation of the above mentioned radial velocity model functions can be found in Chapter 4, in Fig. 4.3.

Extended Markov Chain Monte-Carlo

In this section we discuss in more details one of the built-in methods, that combines a Markov Chain Monte-Carlo algorithm with the parametric derivatives of the model functions in order to yield faster convergence and more reliable results, especially in the cases of highly correlated parameters.

The main concept of the MCMC algorithm (see e.g. Ford, 2004), is to generate an *a posteriori* probability distribution of the adjusted parameters. It is based on random walks in the parameter space as follows. In each step, one draws an alternate parameter vector from an *a priori* distribution and then evaluates the merit function χ^2. If the value of the χ^2 decreases, we accept the transition (since the newly drawn parameter vector represents a better fit), otherwise the transition is accepted by a certain probability (derived from the increment in χ^2). The final distribution of the parameters depends on both the *a priori* distribution and the probability function used when the value of the $\Delta\chi^2$ is positive. The main problem of the MCMC method is that the *a posteriori* probability distribution can only be estimated if the *a priori* distribution is chosen well, but initially we do not have any hint for both distributions. The idea behind MCMC is to derive multiple chains, by taking the *a posteriori* distribution of the previous chain as the input (*a priori*) distribution for the upcoming chain. In regular cases, the chains converge to a final distribution after some iterations and therefore the last one can be accepted as a final result. In the literature, several attempts are known to define an *a priori* transition function (see also Ford, 2004). Here we give a simple method that not only provides a good hint for the *a priori* distribution but yields several independent sanity checks that are then used to verify the convergence of the chain. The transition function used by this extended Markov Chain Monte-Carlo algorithm (XMMC) is a Gaussian distribution of which covariances are derived from the Fisher covariance matrix (Finn, 1992). The sanity checks are then the following:

- The resulted parameter distribution should have nearly the same statistical covariance

as the analytical covariance[49].

- The autocorrelation lengths of the chain parameters have to be small, i.e. nearly $\sim 1-2$ steps. Chains failed to converge have significantly larger autocorrelation lengths.

- The transition probability has to be consistent with the theoretical probabilities. This theoretical probability depends only on the number of adjusted parameters.

- The statistical centroid (mode) of the distribution must agree with both the best fit parameter derived from alternate methods (such as downhill simplex) as well as the chain element with the smallest χ^2.

The method of XMMC has some disadvantages. First, the transition probabilities exponentially decrease as the number of adjusted parameters increases, therefore, the required computational time can be exceptionally high in some cases. The Gibbs sampler (used in the classic MCMC) provides roughly constant transition probability. Second, the derivation of the Fisher covariance matrix requires the knowledge of the parametric derivatives of the merit function. In the actual implementation of `lfit`, XMMC one can use the method of XMMC if the parametric derivatives are known in advance in an analytical form. Otherwise, the XMMC algorithm cannot be applied at all.

However, in the case of HATNet data analysis, we found the method of XMMC to be highly efficient and we used it in several analyses related to the discoveries. Moreover, the most important functions concerning to this analysis, such as light curve and radial velocity model functions have known analytic partial derivatives. These derivatives for transit light curve model functions can be found in Pál (2008). An analytic formalism for radial velocity modelling is discussed in Sec. 4.3 and some additional related details and applications are presented in Pál (2009). In this thesis (in Chapter 3) a detailed example is given on the application of the XMMC algorithm in the analysis of the HAT-P-7(b) planetary system.

2.13   Analysis of photometric data

In this section we describe briefly how the previously discussed algorithms and the respective implementations are used in the practice of photometric data reduction. The concepts for the major steps in the photometry are roughly the same for the HATNet and follow-up data, however, the latter has two characteristics that make the processing more convenient. First, the total amount of frames are definitely smaller, a couple of hundred frames for a singe night or event, while there are thousands or tens of thousands of frames for a typical observation of

---

[49]In practice, the program `lfit` reports the individual uncertainties of the parameters and the correlation matrix. Of course, this information can easily be converted to a covariance matrix and vice versa.

```
# This command just prints the content of the file ``line.dat'' to the standard output:
$ cat line.dat
2  8.10
3 10.90
4 14.05
5 16.95
6 19.90
7 23.10
# Regression:  this command fits a ``straight line'' to the above data:
$ lfit -c x,y -v a,b -f "a*x+b" -y y line.dat
      2.99714        2.01286
# Evaluation:  this command evaluates the model function assuming the parameters to be known:
$ lfit -c x,y -v a=2.99714,b=2.01286 -f "x,y,a*x+b,y-(a*x+b)" -F %6.4g,%8.2f,%8.4f,%8.4f line.dat
        2          8.10      8.0071      0.0929
        3         10.90     11.0043     -0.1042
        4         14.05     14.0014      0.0486
        5         16.95     16.9986     -0.0486
        6         19.90     19.9957     -0.0957
        7         23.10     22.9928      0.1072
$ lfit -c x,y -v a,b -f "a*x+b" -y y line.dat --err
      2.99714        2.01286
    0.0253144      0.121842
```

Figure 2.27: These pieces of commands show the two basic operations of lfit: the first invocation of lfit fits a straight line, i.e. a model function with the form of $ax + b = y$ to the data found in the file line.dat. This file is supposed to contain two columns, one for the x and one for the y values. The second invocation of lfit evaluates the model function. Values for the model parameters (a, b) are taken from the command line while the individual data points (x, y) are still read from the data file line.dat. The evaluation mode allows the user to compute (and print) arbitrary functions of the model parameters *and* the data values. In the above example, the model function itself and the fit residuals are computed and printed, following the read values of x and y. Note that the printed values are formatted for a minimal number significant figures (%6.4g) or for a fixed number of decimals (%8.2f or %8.4f). The last command is roughly the same as the first command for regression, but the individual uncertainties are also estimated by normalizing the value of the χ^2 to unity.

a certain HATNet field. Second, the number of stars on each individual frame is also smaller (a few hundred instead of tens or hundreds of thousands). Third, during the reduction of follow-up photometric data, we have an expectation for the signal shape. The signal can be easily obtained even by lower quality of data and/or when some of the reduction steps are skipped (e.g. trend filtering or a higher order magnitude transformation).

The schematics of a typical photometric pipeline (as used for HATNet data reductions) is shown in Fig. 2.28. It is clear from the figure that the steps of the reduction are the same up to astrometry both in cases when the fluxes are derived either by normal (aperture) photometry or image subtraction method. In the first case, the astrometric solution is directly used to compute the aperture centroids for all objects of interest, while in case of image subtraction, the image registration parameters are based on astrometry. After the instrumental magnitudes are obtained, the process of the photometric files (including transposition, trend filtering and per-object light curve analysis) are the same again. In practice, both primary photometric methods yield fluxes for several apertures. Therefore,
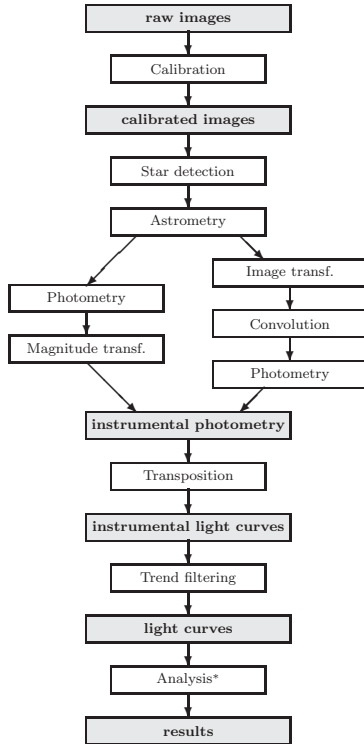
Figure 2.28: Flowchart of the typical photometric reduction pipeline. Each empty box represents a certain step of the data processing that requires non-negligible amount of computing resources. Filled boxes represent the type of data that is only used for further processing, thus the four major steps of the reduction are clearly distinguishable. See text for further details.

joint processing of various photometric data is also feasible since the subsequent steps do not involve additional information beyond the instrumental magnitudes. The only exception is that additional data can be involved in the EPD algorithm in case of image subtraction photometry. Namely, the kernel coefficients $C_{ik\ell}$ can be added to the set of EPD parameters $p^{(i)}$ (see equation 2.84), by evaluating for the spatial variations of each object:

$$p^{(i)} = \sum_{0 \le k+\ell \le N_{\mathrm{K}}^{(i)}} C_{ik\ell} x^k y^\ell, \tag{2.94}$$

where (x, y) is the centroid coordinate of the actual object of interest. In the following two chapters, I discuss how the above outlined techniques are applied in the case of HATNet and follow-up data reductions.

```sh
#!/bin/sh

CATALOG=input.cat   # name of the reference catalog
COLID=1             # column index of object identifie (in the $CATALOG file)
COLX=2              # column index of the projected X coordinate (in the $CATALOG file)
COLY=3              # column index of the projected Y coordinate (in the $CATALOG file)
COLMAG=4            # column index of object magnitude (in the $CATALOG file)
COLCOLOR=5          # column index of object color (in the $CATALOG file)
THRESHOLD=4000      # threshold for star detection
GAIN=4.2            # combined gain of the readout electronics and the A/D converter in electrons/ADU
MAGFLUX=10,10000    # magnitude/flux conversion
APERTURE=5:8:8      # aperture radius, background area inner radius and thickness (all in pixels)


mag_param=c0_00,c0_10,c0_01,c0_20,c0_11,c0_02,c1_00,c1_01,c1_10
mag_funct="c0_00+c0_10*x+c0_01*y+0.5*(c0_20*x^2+2*c0_11*x*y+c0_02*y^2)+color*(c1_00+c1_10*x+c1_01*y)"


for base in ${LIST[*]} ; do
        fistar  ${FITS}/$base.fits --algorithm uplink --prominence 0.0 --model elliptic \
                --flux-threshold $THRESHOLD --format id,x,y,s,d,k,amp,flux -o ${AST}/$base.stars
        grmatch --reference $CATALOG --col-ref $COLX,$COLY --col-ref-ordering -$COLMAG \
                --input ${AST}/$base.stars --col-inp 2,3 --col-inp-ordering +8 \
                --weight reference,column=$COLMAG,magnitude,power=2 \
                --triangulation maxinp=100,maxref=100,conformable,auto,unitarity=0.002 \
                --order 2 --max-distance 1 \
                --comment --output-transformation ${AST}/$base.trans || continue
        grtrans $CATALOG --col-xy $COLX,$COLY --col-out $COLX,$COLY \
                --input-transformation ${AST}/$base.trans --output - | \
        fiphot  ${FITS}/$base.fits --input-list - --col-xy $COLX,$COLY --col-id $COLID \
                --gain $GAIN --mag-flux $MAGFLUX --aperture $APERTURE --disjoint-annuli \
                --sky-fit mode,iterations=4,sigma=3 --format IXY,MmBbS \
                --comment --output ${PHOT}/$base.phot
        paste   ${PHOT}/$base.phot ${PHOT}/$REF.phot $CATALOG | \
        lfit    --columns mag:4,err:5,mag0:12,x:10,y:11,color:$((2*8+COLCOLOR)) \
                --variables $mag_param --function "$mag_funct" --dependent mag0-mag --error err \
                --output-variables ${PHOT}/$base.coeff
        paste   ${PHOT}/$base.phot ${PHOT}/$REF.phot | \
        lfit    --columns mag:4,err:5,mag0:12,x:10,y:11,color:$((2*8+COLCOLOR)) \
                --variables $(cat ${PHOT}/$base.coeff) \
                --function "mag+($mag_funct)" --format %9.5f --column-output 4 | \
        awk     '{ print $1,$2,$3,$4,$5,$6,$7,$8; }' > ${PHOT}/$base.tphot
done
for base in ${LIST[*]} ; do test -f ${PHOT}/$base.tphot && cat ${PHOT}/$base.tphot ; done | \
grcollect - --col-base 1 --prefix $LC/ --extension .lc
```

Figure 2.29: A shell script demonstrating a complete working pipeline for aperture photometry. The input FITS files are read from the directory ${FITS} and their base names (without the *.fits extension) are supposed to be listed in the array ${LIST[*]}. These base names are then used to name the files storing data obtained during the reduction process. Files created by the subsequent calls of the fistar and grmatch programs are related to the derivation of the astrometric solution and the respective files are stored in the directory ${AST}. The photometry centroids are derived from the original input catalog (found in the file $CATALOG) and the astrometric transformation (plate solution, stored in the *.trans files). The results of the photometry are put into the directory ${PHOT}. Raw photometry is followed by the magnitude transformation. This branch involves additional common UNIX utilities such as paste and awk in order to match the current and the reference photometry as well as to filter and resort the output after the magnitude transformation. The derivation of the transformation coefficients is done by the lfit utility, that involves $mag_funct with the parameters listed in $mag_param. This example features a quadratic magnitude transformation and a linear color dependent correction (to cancel the effects of the differential refraction). The final light curves are created by the grcollect utility what writes the individual files into the directory ${LC}.

Chapter 3

HATNet discoveries

In the past few years, the HATNet project announced 11 discoveries and became one of the most successful initiatives searching for transiting extrasolar planets. In this chapter the procedures of the photometric measurements and analysis of spectroscopic data (including radial velocity are explained, emphasizing how the algorithms and programs were used in the data reduction and analysis. The particular example of the planetary system HAT-P-7(b) clearly demonstrates all of the necessary steps that are generally required by the detection and confirmation of transiting extrasolar planets. In Sec. 3.1, the issues related to the primary photometric detection are explained. Sec 3.2 summarizes the follow-up observations, which are needed by the proper confirmation of the planetary nature. Mainly, the roles of these photometric follow-up observations are treefold. First, it provides additional data in order to have a better estimation of the planetary parameters whose are derived from the light curve of the system. Like so, spectroscopic analysis yields additional information from which the planetary mass or the properties and physical parameters of the host star can be deduced. Third, analysis of follow-up data helps to exclude other scenarios that are likely to show similar photometric or spectroscopic variations what a transiting extrasolar planet shows. In Sec 3.3, the methods are explained that we were using to obtain the final planetary parameters.

3.1 Photometric detection

The HATNet telescopes HAT-7 and HAT-8 (HATNet; Bakos et al., 2002, 2004) observed HATNet field G154, centered at $\alpha = 19^{\mathrm{h}}12^{\mathrm{m}}$, $\delta = +45°00'$, on a near-nightly basis from 2004 May 27 to 2004 August 6. Exposures of 5 minutes were obtained at a 5.5-minute cadence whenever conditions permitted; all in all 5140 exposures were secured, each yielding photometric measurements for approximately $33,000$ stars in the field down to $I \sim 13.0$. The field was observed in network mode, exploiting the longitude separation between HAT-7,
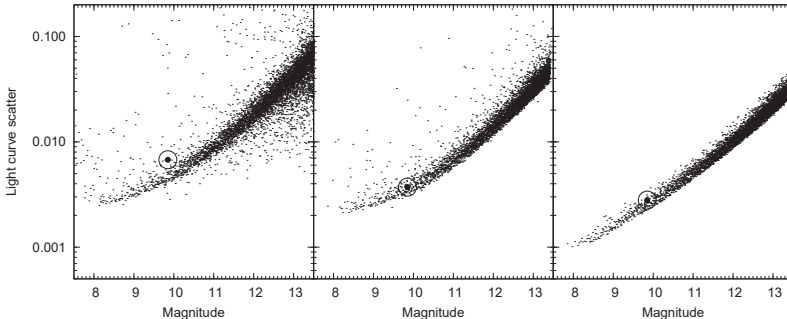
Figure 3.1: Light curve statistics for the field "G154", obtained by aperture photometry (left panel) and photometry based on the method of image subtraction (middle panel). The right panel shows the lower noise limit estimation derived from the Poisson- and background noise. Due to the strong vignetting of the optics, the effective gain varies across the image. Therefore, the distribution of the points on the right panel is not a clear thin line. Instead, the thickness of the line is approximately equivalent to a factor of ~ 2 between the noise level, indicating a highly varying vignetting of a factor of ~ 4. The star HAT-P-7 (GSC 03547-01402) is represented by the thick dot. The light curve scatter for this star has been obtained involving only out-of-transit data. This star is a prominent example where the method of image subtraction photometry significantly improves the light curve quality.

stationed at the Smithsonian Astrophysical Observatory's (SAO) Fred Lawrence Whipple Observatory (FLWO) in Arizona ($\lambda = 111°\,\text{W}$), and HAT-8, installed on the rooftop of SAO's Submillimeter Array (SMA) building atop Mauna Kea, Hawaii ($\lambda = 155°\,\text{W}$). We note that each light curve obtained by a given instrument was shifted to have a median value to be the same as catalogue magnitude of the appropriate star, allowing to merge light curves acquired by different stations and/or detectors.

Following standard frame calibration procedures, astrometry was performed as described in Sec. 2.5, and aperture photometry results (see Sec. 2.7 and Sec. 2.12.13) were subjected to External Parameter Decorrelation (EPD, Sec. 2.10), and also to the Trend Filtering Algorithm ((TFA; see Sec. 2.10 or Kovács, Bakos & Noyes, 2005). We searched the light curves of field G154 for box-shaped transit signals using the BLS algorithm of Kovács, Zucker & Mazeh (2002). A very significant periodic dip in brightness was detected in the $I \approx 9.85$ magnitude star GSC 03547-01402 (also known as 2MASS 19285935+4758102; $\alpha = 19^\text{h}28^\text{m}59^\text{s}.35$, $\delta = +47°58'10''.2$; J2000), with a depth of $\sim 7.0\,\text{mmag}$, a period of $P = 2.2047$ days and a relative duration (first to last contact) of $q \approx 0.078$, equivalent to a duration of $Pq \approx 4.1$ hours.

In addition, the star happened to fall in the overlapping area between fields G154 and G155. Field G155, centered at $\alpha = 19^\text{h}48^\text{m}$, $\delta = +45°00'$, was also observed over an extended time in between 2004 July 27 and 2005 September 20 by the HAT-6 (Arizona) and HAT-9 (Hawaii) telescopes. We gathered 1220 and 10260 data-points, respectively (which independently confirmed the transit), yielding a total number of 16620 data-points.
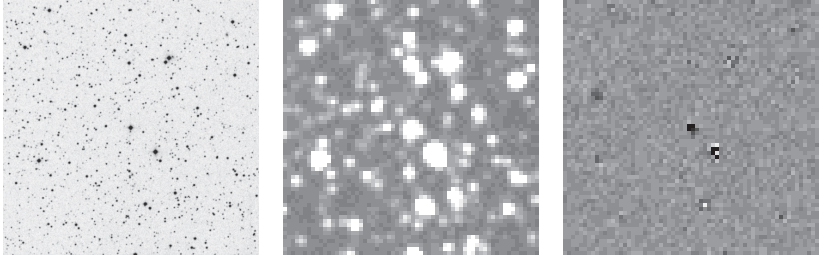
Figure 3.2: Stamps showing the vicinity of the star HAT-P-7. All of the stamps have the same size, covering an area of $15.7' \times 15.7'$ on the sky and centered on HAT-P-7. The left panel is taken from the POSS-1 survey (available, e.g. from the STScI Digitized Sky Survey web page). The middle panel shows the same area, as the HATNet telescopes see it. This stamp was cut from the photometric reference image (as it was used for the image subtraction process), that was derived from the ~ 20 sharpest and cleanest images of the HAT-8 telescope. The right panel shows the convolution residual images averaged on the ~ 160 frames acquired by the HAT-8 telescope during the transit. The small dip at the center of the image can be seen well. Some residual structures at the positions of brighter stars also present.

After the announcement and the publication of the planet HAT-P-7b (Pál et al., 2008a), all of the images for the fields G154 and G155 were re-analyzed by the method of image subtraction photometry. Based on the astrometric solution[1], the images were registered to the coordinate system of one of the images that was found to be a proper reference image (Sec. 2.6). From the set of registered frames approximately a dozen of them have been chosen to create a good signal-to-noise ratio master reference image for the image subtraction procedure. These frames were selected to be the sharpest ones, i.e. where the overall profile sharpness parameter, S (see Sec. 2.4.2) were the largest among the images (note that large S corresponds to small FWHM, i.e. to sharp stars). Moreover, such images were chosen from the ones where the Moon was below the horizon (see also Fig. 2.15 and the related discussion). The procedure was repeated for both fields G154 and G155. The intensity levels of these individual sharp frames were then transformed to the same level involving the program `ficonv`, with a formal kernel size of 1×1 pixels ($B_K = 0$, $N_{kernel} = 1$, $K^{(1)} = \delta^{(00)}$). Such an intensity level transformation corrects for the changes in the instrumental stellar brightnesses due to the varying airmass, transparency and background level. These images were then combined (Sec. 2.12.4) in order to have a single master convolution reference image. This step was performed for both of the fields. The reference images were then used to derive the optimal convolution transformation, and simultaneously the residual ("subtracted") images were also obtained by `ficonv`. For each individual object image, both the result of the convolution kernel fit and the residual image were saved to files for further processing. For the fit, we have employed a discrete kernel basis with the size of 7×7 pixels and we let a spatial variation of 4th polynomial order for both the kernel parameters and the background

---

[1]The astrometric solutions have been already obtained at this point since the source identification and the centroid coordinates were already required earlier by aperture photometry.

level. Due to the sharp profiles (the profile FWHMs were between $2.0 \ldots 2.4$), this relatively small kernel size were sufficient for our purposes. The residuals on the subtracted images were subjected to aperture photometry, based on the considerations discussed in Sec. 2.9. For the proper image subtraction-based photometry, one needs to derive and to use the fluxes on the reference image as well. These fluxes were derived using aperture photometry, and the instrumental raw magnitudes were transformed to the catalogue magnitudes with a fourth order polynomial transformation. The residual of this fit was nearly 0.05 mags for both fields, thus the fluxes of the individual stars have been well determined, and this transformation yielded proper reference fluxes even for the faint and the blended stars. The results of the image subtraction photometry were then processed similarly to the normal aperture photometry results (see also Fig. 2.28), and the respective light curves were de-trended involving both the EPD and TFA algorithms.

For a comparison, the light curve residuals for the normal aperture photometry and the image subtraction photometry are plotted on the left and middle panel of Fig. 3.1. In general, the image subtraction photometry yielded light curve residuals smaller by a factor of $\sim 1.2 - 1.5$. The gain achieved by the image subtraction photometry is larger for the fainter stars. It is important to note that in the case of the star HAT-P-7, the image subtraction photometry improved the photometric quality[2] by a factor of ~ 1.8: the rms of the out-of-transit section in the aperture photometry light curve were 6.75 mmag while the image subtraction method yielded an rms of 3.72 mmag. The lower limit of the intrinsic noise of this particular star is 2.8 mmag (see also the right panel of Fig. 3.1). In Fig. 3.2, we display some image stamps from the star HAT-P-7 and its neighborhood. Since the dip of ~ 7 mmag during the transits of HAT-P-7b is only ~ 2 times larger than the overall rms of the light curve, individual subtracted frames does not significantly show the "hole" at the centroid position of the star, especially because this weak signal is distributed among several pixels. Therefore, on the right panel of Fig. 3.2, all of the frames acquired by the telescope HAT-8 during the transit have been averaged in order to show a clear visual detection of the transit. Albeit the star HAT-P-7 is a well isolated one, such visual analysis of image residuals can be relevant when the signal is detected for stars whose profiles are significantly merged. In such cases, either the visual analysis or a more precise quantification of this "negative residual" (e.g. by employing the star detection and characterization algorithms of Sec. 2.4) can help to distinguish which star is the variable.

The combined HATNet light curve, yielded by the image subtraction photometry and de-trended by the EPD and TFA is plotted on Fig. 3.3. Superimposed on these plots is our best fit model (see Sec. 3.3). We note that TFA was run in signal reconstruction mode, i.e. systematics were iteratively filtered out from the observed time series assuming that

---

[2]In the case of a star having periodic dips in its light curve, the scatter is derived only from the out-of-transit sections.

the underlying signal is a trapeze-shaped transit (see Sec. 2.10 and Kovács, Bakos & Noyes, 2005, for additional details). We note that fields G154 and G155 both intersect the field of view of the Kepler mission (Borucki et al., 2007), and more importantly, HAT-P-7 lies in the Kepler field.

3.2 Follow-up observations

3.2.1 Reconnaissance spectroscopy

Following the HATNet photometric detection, HAT-P-7 (then a transit *candidate*) was observed spectroscopically with the CfA Digital Speedometer (DS, see Latham, 1992) at the FLWO 1.5 m Tillinghast reflector, in order to rule out a number of blend scenarios that mimic planetary transits (e.g. Brown, 2003; O'Donovan et al., 2007), as well as to characterize the stellar parameters, such as surface gravity, effective temperature, and rotation. Four spectra were obtained over an interval of 29 days. These observations cover 45 Å in a single echelle order centered at 5187 Å, and have a resolving power of $\lambda/\Delta\lambda \approx 35{,}000$. Radial velocities were derived by cross-correlation, and have a typical precision of $1\,\mathrm{km\,s^{-1}}$. Using these measurements, together with collaborators, we have ruled out an unblended companion of stellar mass (e.g. an M dwarf orbiting an F dwarf), since the radial velocities did not show any variation within the uncertainties. The mean heliocentric radial velocity of HAT-P-7 was measured to be $-11\,\mathrm{km\,s^{-1}}$. Based on an analysis similar to that described in Torres et al. (2002), the DS spectra indicated that the host star is a slightly evolved dwarf with $\log g = 3.5$ (cgs), $T_{\mathrm{eff}} = 6250\,\mathrm{K}$ and $v\sin i \approx 6\,\mathrm{km\,s^{-1}}$.

3.2.2 High resolution spectroscopy

For the characterization of the radial velocity variations and for the more precise determination of the stellar parameters, we obtained 8 exposures with an iodine cell, plus one iodine-free template, using the HIRES instrument (Vogt et al., 1994) on the Keck I telescope, Hawaii, between 2007 August 24 and 2007 September 1. The width of the spectrometer slit was $0''.86$ resulting a resolving power of $\lambda/\Delta\lambda \approx 55{,}000$, while the wavelength coverage was $\sim 3800 - 8000$ Å. The iodine gas absorption cell was used to superimpose a dense forest of I_2 lines on the stellar spectrum and establish an accurate wavelength fiducial (see Marcy & Butler, 1992). Relative radial velocities in the Solar System barycentric frame were derived as described by Butler et al. (1996), incorporating full modeling of the spatial and temporal variations of the instrumental profile. The final radial velocity data and their errors are listed in Table 3.1. The folded data, with our best fit (see Sec. 3.3.2) superimposed, are plotted in Fig. 3.7a.
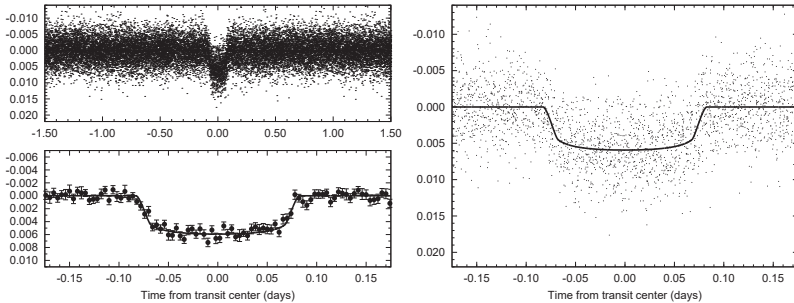
Figure 3.3: Upper left panel: the complete light curve of HAT-P-7 with all of the 16620 points, unbinned instrumental I-band photometry obtained with four telescopes of HATNet (see text for details), and folded with the period of $P = 2.2047298$ days (the result of a joint fit to all available data, Sec. 3.3.2). The superimposed curve shows the best model fit using quadratic limb darkening. Right panel: The transit zoomed-in (3150 data points are shown). Lower left panel: same as the right panel, with the points binned with a bin size of 0.004 in days.
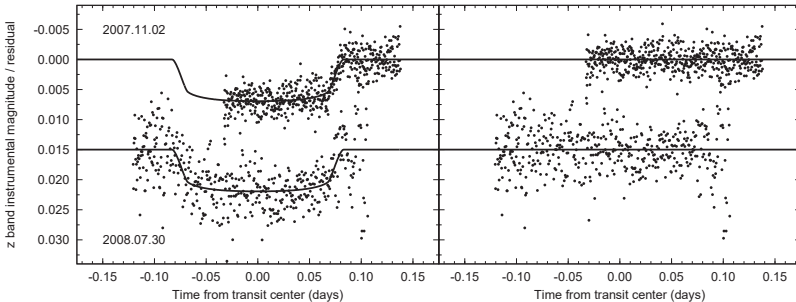
Figure 3.4: Left panel: unbinned instrumental Sloan z-band partial transit photometry acquired by the KeplerCam at the FLWO 1.2 m telescope on 2007 November 2 and 2008 July 30; superimposed is the best-fit transit model light curve. Right panel: the difference between the KeplerCam observation and model (on the same vertical scale).

3.2.3 Photometric follow-up observations

Partial photometric coverage of a transit event of HAT-P-7 was carried out in the Sloan z-band with the KeplerCam CCD on the 1.2 m telescope at FLWO, on 2007 November 2. The total number of frames taken from HAT-P-7 was 514 with cadence of 28 seconds. During the reduction of the KeplerCam data, we used the following method. After bias and flat calibration of the images, an astrometric transformation (in the form of first order polynomials) between the ~ 450 brightest stars and the 2MASS catalog was derived, as described in Sec. 2.5, yielding a residual of $\sim 0.2 - 0.3$ pixel. Aperture photometry was then performed using a series of apertures with the radius of 4, 6 and 8 pixels in fixed positions

calculated from this solution and the actual 2MASS positions. The instrumental magnitude transformation was obtained using ~ 350 stars on a frame taken near culmination of the field. The transformation fit was initially weighted by the estimated photon- and background-noise error of each star, then the procedure was repeated by weighting with the inverse variance of the light curves. From the set of apertures we have chosen the aperture for which the out-of-transit (OOT) rms of HAT-P-7 was the smallest; the radius of this aperture is 6 pixels. The resulted light curve has been presented in the discovery paper of Pál et al. (2008a). More recently, in 2008 July 30, we have obtained an additional complete light curve for the transit of HAT-P-7b, also in Sloan z-band with the KeplerCam CCD.

The two follow-up light curves from 2007 November 2 and 2008 July 30 were then decorrelated against trends using the complete data, involving a simultaneous fit for the light curve model function parameters and the EPD parameters (see also Sec. 3.3). These fits yielded a light curve with an overall rms of 1.83 mmag and 4.23 mmag for these two nights, respectively. In both cases, the cadence of the individual photometric measurements were 28 seconds. For the first night the residual scatter of 1.83 mmag is a bit larger than the expected rms of 1.5mmag, derived from the photon noise (1.2mmag) and scintillation noise – that has an expected amplitude of 0.8mmag, based on the observational conditions and the calculations of Young (1967) – possibly due to unresolved trends and other noise sources. For the second night, the photometric quality was significantly worse, due to the high variations in the transparency[3]. The resulting light curves are shown in Fig. 3.4, superimposed with our best fit model (Sec. 3.3).

3.2.4 Excluding blend scenarios

Following Torres et al. (2007), we explored the possibility that the measured radial velocities are not real, but instead caused by distortions in the spectral line profiles due to contamination from a nearby unresolved eclipsing binary. In that case the "bisector span" of the average spectral line should vary periodically with amplitude and phase similar to the measured velocities themselves (Queloz et al., 2001; Mandushev et al., 2005). We cross-correlated each Keck spectrum against a synthetic template matching the properties of the star (i.e. based on the SME results, see Sec. 3.3.4), and averaged the correlation functions over all orders blueward of the region affected by the iodine lines. From this representation of the average spectral line profile we computed the mean bisectors, and as a measure of the line asymmetry we computed the "bisector spans" as the velocity difference between points selected near the top and bottom of the mean bisectors (Torres et al., 2005). If the velocities were the result of a blend with an eclipsing binary, we would expect the line bisectors to vary

---

[3]For 2007 November 2, the scatter of the *raw* magnitudes were ~ 14 mmag while on the night of 2008 July 30, the raw magnitude rms were more than 15 times higher, nearly 0.24 mag.

in phase with the photometric period with an amplitude similar to that of the velocities. Instead, we detect no variation in excess of the measurement uncertainties (see Fig. 3.7c). We have also tested the significance of the correlation between the radial velocity and the bisector variations. Therefore, we conclude that the velocity variations are real and that the star is orbited by a Jovian planet. We note here that the mean bisector span ratio relative to the radial velocity amplitude is the smallest (~ 0.026) among all the HATNet planets, indicating an exceptionally high confidence that the RV signal is not due to a blend with an eclipsing binary companion.

3.3   Analysis

The analysis of the available data was done in four steps. First, an independent analysis was performed on the HATNet, the radial velocity (RV) and the high precision photometric follow-up (FU) data, respectively. Analysis of the HATNet data yielded an initial value for the orbital period and transit epoch. The initial period and epoch were used to fold the RV's, and phase them with respect to the predicted transit time for a circular orbit. The HATNet and the RV epochs together yield a more accurate period, since the time difference between the discovery light curve and the RV follow-up is fairly long; more than 3 years. Using this refined period, we can extrapolate to the expected center of the KeplerCam partial transit, and therefore obtain a fit for the two remaining key parameters describing the light curve: $a/R_\star$ where a is the semi-major axis for a circular orbit, and the impact parameter $b \equiv (a/R_\star)\cos i$, where i is the inclination of the orbit.

Second, using as starting points the initial values as derived above, we performed a joint fit of the HATNet, RV and FU data, i.e. fitting *all* of the parameters simultaneously. The reason for such a joint fit is that the three separate data-sets and the fitted parameters are intertwined. For example, the epoch (depending partly on the RV fit) has a relatively large error, affecting the extrapolation of the transit center to the KeplerCam follow-up.

In the discovery report, in all of the above procedures, we used the downhill simplex method (DHSX, Sec. 2.12.16) to search for the best fit values and the method of refitting to synthetic data sets (called EMCE, see also Sec. 2.12.16) to find out the error of the adjusted parameters. The refined analysis based on the HATNet light curves reduced by the method of image subtraction photometry and an additional photometric measurement from the night of 2008 July 30 was also involved. In this analysis the extended Markov Chain Monte-Carlo algorithm (XMMC) was employed, also in the form of an implementation found in the program lfit. As it was mentioned in Sec. 2.12.16, the XMMC method used in this particular analysis has also been aided by the DHSX minimization (as a first iteration) and used as a sanity check of the chain convergence (see also Sec. 3.3). Both of these

Table 3.1: Relative radial velocity (RV) and bisector span (BS) measurements of HAT-P-7. The RV and BS data points, as well as their formal errors are given in units of m/s.

BJD	RV	σ_{RV}	BS	σ_{BS}
2454336.73121	-	-	5.30	5.36
2454336.73958	+124.40	1.63	0.68	5.10
2454336.85366	+73.33	1.48	4.82	6.17
2454337.76211	−223.89	1.60	−1.94	5.30
2454338.77439	+166.71	1.39	2.58	5.35
2454338.85455	+144.67	1.42	7.60	5.22
2454339.89886	−241.02	1.46	−5.13	5.77
2454343.83180	−145.42	1.66	−8.30	6.58
2454344.98804	+101.05	1.91	−5.62	5.80

error estimation methods (EMCE and XMMC) yield a Monte-Carlo set of the *a posteriori* distribution of the fit parameters, that were subsequently used in the derivation of the final planetary, orbital and stellar characteristics.

The third step of the analysis was the derivation of the stellar parameters, based on the spectroscopic analysis of the host star (high resolution spectroscopy using Keck/HIRES), and the physical modeling of the stellar evolution, based on existing isochrone models. As the fourth step, we then combined the results of the joint fit and stellar parameter determination to determine the planetary and orbital parameters of the HAT-P-7b system. In the following we summarize these steps.

3.3.1 Independent fits

For the independent fit procedure, we first analyzed the HATNet light curves, as observed by the HAT-6, HAT-7, HAT-8 and HAT-9 telescopes. Using the initial period and transit length from the BLS analysis, we fitted a model to the 214 cycles of observations spanned by all the HATNet data. Although at this stage we were interested only in the epoch and period, we have used the transit light curve model with the assumption of quadratic limb darkening, where the flux decrease was calculated using the models provided by Mandel & Agol (2002). In principle, fitting the epoch and period as two independent variables is equivalent to fitting the time instant of the centers of the first and last observed individual transits, $T_{\mathrm{c,first}}$ and $T_{\mathrm{c,last}}$, with a constraint that all intermediate transits are regularly spaced with period P. Note that this fit takes into account *all* transits that occurred during the HATNet observations, even though it is described only by $T_{\mathrm{c,first}}$ and $T_{\mathrm{c,last}}$. The fit yielded $T_{\mathrm{c,first}} = 2453153.0924 \pm 0.0021$ (BJD) and $T_{\mathrm{c,last}} = 2453624.9044 \pm 0.0023$ (BJD). the correlation between these two epochs turned out to be: $C(T_{\mathrm{c,first}}, T_{\mathrm{c,last}}) = -0.53$. The period derived

from the $T_{\mathrm{c,first}}$ and $T_{\mathrm{c,last}}$ epochs was $P^{(1)} = 2.20480 \pm 0.00049$ days. Using these values, we found that there were 326 cycles between $T_{\mathrm{c,last}}$ and the end of the RV campaign. The epoch extrapolated to the approximate time of RV measurements was $T_{\mathrm{c,RV}} = 2454343.646 \pm 0.008$ (BJD). Note that the error in $T_{\mathrm{c,RV}}$ is much smaller than the period itself (~ 2.2 days), so there is no ambiguity in the number of elapsed cycles when folding the periodic signal.

We then analyzed the radial velocity data in the following way. We defined the $N_{\mathrm{tr}} \equiv 0$ transit as that being closest to the end of the radial velocity measurements. This means that the first transit observed by HATNet (at $T_{\mathrm{c,first}}$) was the $N_{\mathrm{tr,first}} = -569$ event. Given the short period, we assumed that the orbit has been circularized (Hut, 1981) (later verified; see below). The orbital fit is linear if we choose the radial velocity zero-point γ and the amplitudes A and B as adjusted values, namely:

$$v(t) = \gamma + A \cos\left[\frac{2\pi}{P}(t - t_0)\right] + B \sin\left[\frac{2\pi}{P}(t - t_0)\right], \tag{3.1}$$

where t_0 is an arbitrary time instant (chosen to be $t_0 = 2454342.6$ BJD), $K \equiv \sqrt{A^2 + B^2}$ is the semi-amplitude of the RV variations, and P is the initial period $P^{(1)}$ taken from the previous independent HATNet fit. The actual epoch can be derived from the above equation, since for circular orbits the transit center occurs when the RV curve has the most negative slope. For circular orbits, the transit occurs at the time instant when the RV curve has the smallest time-derivative, the actual epoch of the transit must be:

$$T_c = t_0 + \frac{P}{2\pi} \arg(-B, A) = t_0 + \frac{P}{2\pi} \arctan\left(-\frac{A}{B}\right). \tag{3.2}$$

Using the equations above, we derived the initial epoch of the $N_{\mathrm{tr}} = 0$ transit center to be $T_c = 2454343.6462 \pm 0.0042 \equiv T_{\mathrm{c,-29}}^{(1)}$ (BJD). We also performed a more general (non-linear) fit to the RV in which we let the eccentricity float. This fit yielded an eccentricity consistent with zero, namely $e \cos \omega = -0.003 \pm 0.007$ and $e \sin \omega = 0.000 \pm 0.010$. Therefore, we adopt a circular orbit in the further analysis.

Combining the RV epoch $T_{\mathrm{c,-29}}^{(1)}$ with the first epoch observed by HATNet ($T_{\mathrm{c,first}}$), we obtained a somewhat refined period, $P^{(2)} = 2.204732 \pm 0.000016$ days. This was fed back into phasing the RV data, and we performed the RV fit again to the parameters γ, A and B. The fit yielded $\gamma = -37.0 \pm 1.5 \, \mathrm{m\,s^{-1}}$, $K \equiv \sqrt{A^2 + B^2} = 213.4 \pm 2.0 \, \mathrm{m\,s^{-1}}$ and $T_{\mathrm{c,-29}}^{(2)} = 2454343.6470 \pm 0.0042$ (BJD). This epoch was used to further refine the period to get $P^{(3)} = 2.204731 \pm 0.000016 \, \mathrm{d}$, where the error calculation assumes that $T_{\mathrm{c,-29}}$ and $T_{\mathrm{c,-569}}$ are uncorrelated. At this point we stopped the above iterative procedure of refining the epoch and period; instead a final refinement of epoch and period was obtained through performing a joint fit, (as described later in Sec. 3.3.2). We note that in order to get a reduced chi-square value near unity for the radial velocity fit, it was necessary to quadratically increase the noise component with an amplitude of $3.8 \, \mathrm{m\,s^{-1}}$, which is well within the range of stellar jitter observed for late F stars; see Butler et al. (2006).

```
# Downhill simplex best fit value:
2453153.09286 2454678.76582   213.35 ... 0.07619 0.2061 13.4529 ... 2759.01432
# XMMC values:
2453153.09356 2454678.76621   211.99 ... 0.07478 0.1864 13.4551 ... 2766.28256
2453153.09446 2454678.76562   213.75 ... 0.07625 0.3036 13.6665 ... 2768.77404
2453153.09472 2454678.76638   213.41 ... 0.07616 0.0714 13.6635 ... 2769.49600
2453153.09473 2454678.76449   213.61 ... 0.07489 0.0190 13.2588 ... 2766.48611
2453153.09468 2454678.76509   215.68 ... 0.07541 0.0542 13.2887 ... 2769.26264
2453153.09496 2454678.76499   214.70 ... 0.07685 0.1223 13.3642 ... 2767.24865
2453153.09465 2454678.76477   214.68 ... 0.07736 0.1542 13.2951 ... 2767.86983
2453153.09474 2454678.76425   214.25 ... 0.07708 0.3420 13.3437 ... 2768.80460
2453153.09371 2454678.76438   213.48 ... 0.07622 0.3270 13.3973 ... 2766.88065
2453153.09358 2454678.76455   216.24 ... 0.07376 0.0295 13.3136 ... 2767.79127
2453153.09317 2454678.76618   211.07 ... 0.07474 0.2070 13.6488 ... 2765.08235
.....................................................................
#
# Accepted transitions / total iterations:  4000/30299
# Total acceptance ratio :  0.13202 +/- 0.00209
# Theoretical probability:  0.14493 [independent:10=23-13-0 (total-constrained-linear)]
#
# Correlation lengths:
# 2.64 2.38 1.80 ....  1.47 0.83 2.20 ...
#
# chi^2 values:
# minimal:  0.928961
# Appropriate values for this chi^2:
# 2453153.09286 2454678.76582 213.35 ... 0.07619 0.2061 13.4529 ...
#
# Errors and correlations (projected Fisher matrix):
# 0.00085 0.00110 1.92 ...  0.00132 0.1272 0.1906
#
#  1.000 -0.195 -0.002  ...  -0.013 -0.023 -0.002
# -0.195  1.000 -0.064  ...   0.043  0.047  0.683
# -0.002 -0.064  1.000  ...  -0.003  0.003 -0.046
# ............................................
# -0.013  0.043 -0.003  ...   1.000  0.728  0.256
# -0.023  0.047 -0.003  ...   0.728  1.000  0.389
# -0.002  0.683 -0.046  ...   0.256  0.389  1.000
#
# Errors and correlations (statistical, around the best fit):
# 0.00109 0.00134 2.04 ...  0.00118 0.1050 0.2221
#
#  1.000 -0.149 -0.015 ...   0.017  0.018  0.018
# -0.149  1.000 -0.126 ...   0.039  0.102  0.796
# -0.015 -0.126  1.000 ...  -0.025 -0.037 -0.086
# ............................................
#  0.017  0.039 -0.025 ...   1.000  0.554  0.124
#  0.018  0.102 -0.037 ...   0.554  1.000  0.318
#  0.018  0.796 -0.086 ...   0.124  0.318  1.000
```

Figure 3.5: The output of the program lfit showing the results of the extended Markov Chain Monte-Carlo (XMMC) analysis related to the HAT-P-7(b) planetary system. The parameters in the output are T_{-569}, T_{+123}, K, $R_{\mathrm{p}}/R_{\star}$, b^2 and $\zeta/R_{\star}$, respectively. For clarity, the other parameters were cut from the output list.
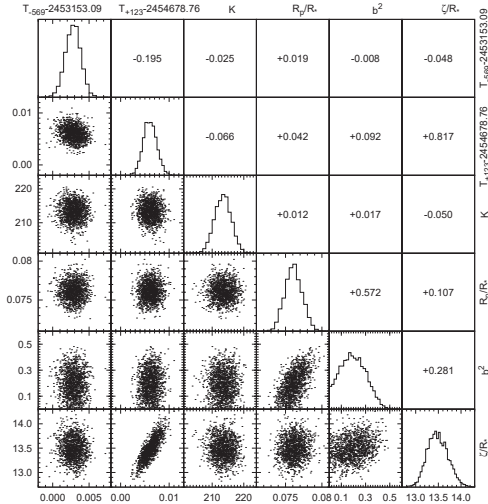
Figure 3.6: Probability distributions and mutual correlations of the adjusted parameters T_{-569}, T_{+123}, K, $R_p/R_\star$, b^2 and $\zeta/R_\star$ for the planet HAT-P-7b. These are the only adjusted parameters of the analysis that are explicitly related to the physical properties of the planet and its orbit. The derivation of these distributions were performed exploiting the extended Markov Chain Monte-Carlo (XMMC) algorithm as it is implemented in the program `lfit` (the related output is shown partially in Fig. 3.5). See text for further details.

Using the improved period $P^{(3)}$ and the epoch $T_{c,-29}$, we extrapolated to the center of KeplerCam follow-up transit ($N_{tr} = 29$). Since the follow-up observation only recorded a partial event (see Fig. 3.4), this extrapolation was necessary to improve the light curve modeling. For this, we have used a quadratic limb-darkening approximation, based on the formalism of Mandel & Agol (2002). The limb-darkening coefficients were based on the results of the SME analysis (notably, T_{eff}; see Sec. 3.3.4 for further details), which yielded $\gamma_1^{(z)} = 0.1329$ and $\gamma_2^{(z)} = 0.3738$. Using these values and the extrapolated time of the transit center, we adjusted the light curve parameters: the relative radius of the planet $p = R_p/R_\star$, the square of the impact parameter b^2 and the quantity $\zeta/R_\star = (a/R_\star)(2\pi/P)(1-b^2)^{-1/2}$ as independent parameters (see Bakos et al., 2007c, for the choice of parameters). The result of the fit was $p = 0.0762 \pm 0.0012$, $b^2 = 0.205 \pm 0.144$ and $\zeta/R_\star = 13.60 \pm 0.83\,\mathrm{day}^{-1}$, where the uncertainty of the transit center time due to the relatively high error in the transit epoch $T_{c,-29}$ was also taken into account in the error estimates.
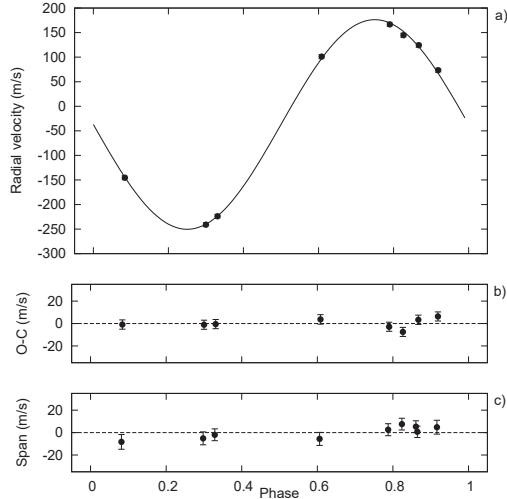
Figure 3.7: (a) Radial-velocity measurements from Keck for HAT-P-7, along with an orbital fit, shown as a function of orbital phase, using our best fit as period (see Sec. 3.3.2). The center-of-mass velocity has been subtracted. (b) Phased residuals after subtracting the orbital fit (also see Sec. 3.3.2). The rms variation of the residuals is about $3.8\,\mathrm{m\,s^{-1}}$. (c) Bisector spans (BS) for the 8 Keck spectra plus the single template spectrum, computed as described in the text. The mean value has been subtracted. Due to the relatively small errors comparing to the RV amplitude, the vertical scale on the (b) and (c) panels differ from the scale used on the top panel.

3.3.2 Joint fit based on the aperture photometry data and the single partial follow-up light curve

The results of the individual fits described above provide the starting values for a joint fit, i.e. a simultaneous fit to all of the available HATNet, radial velocity and the partial follow-up light curve data. The adjusted parameters were $T_{c,-569}$, the time of first transit center in the HATNet campaign, m, the out-of-transit magnitude of the HATNet light curve in I-band and the previously defined parameters of γ, A, B, p, b^2 and $\zeta/R_\star$. We note that in this joint fit *all* of the transits in the HATNet light curve have been adjusted simultaneously, tied together by the constraint of assuming a strictly periodic signal; the shape of all these transits were characterized by p, b^2 and $\zeta/R_\star$ (and the limb-darkening coefficients) while the distinct transit center time instants were interpolated using $T_{c,-569} = T_{c,\mathrm{first}}$ and A, B via the RV fit. For initial values we used the results of the independent fits (Sec. 3.3.1). The error estimation based on method refitting to synthetic data sets gives the distribution of the adjusted values, and moreover, this distribution can be used directly as an input for a Monte-Carlo parameter determination for stellar evolution modeling, as described later (Sec. 3.3.4).

Final results of the joint fit were: $T_{c,-569} = 2453153.0924 \pm 0.0015$ (BJD), $m = 9.85053 \pm 0.00015$ mag, $\gamma = -37.0 \pm 1.5\,\mathrm{m\,s^{-1}}$, $A = 33.8 \pm 0.9\,\mathrm{m\,s^{-1}}$, $B = 210.7 \pm 1.9\,\mathrm{m\,s^{-1}}$, $p = 0.0763 \pm 0.0010$, $b^2 = 0.135^{+0.149}_{-0.116}$ and $\zeta/R_\star = 13.34 \pm 0.23$ day^{-1}. Using the distribution of these parameters, it is straightforward to obtain the values and the errors of the additional parameters derived from the joint derived fit, namely $T_{c,-29}$, $a/R_\star$, K and P. All final fit parameters are listed in Table 3.3.

3.3.3 Joint fit based on the image subtraction photometry data and both of the follow-up light curves

Involving the additional recent follow-up photometry data from 2008 July 30 and the HAT-Net light curve obtained by the method based on image subtraction, we repeated the analysis of the available data. In this new analysis, the method of extended Markov Chain Monte-Carlo (XMMC) has been employed to derive the best fit parameters and their *a posteriori* distributions. Due to the presence of a complete photometric follow-up light curve, we have used a slightly different set of parameters. Moreover, the trend filtering based on the EPD algorithm has been performed simultaneously with the fit. Thus, the set of adjusted parameters that are related to the physical properties of the planetary system were the following: the center of the first transit measured by the HATNet telescopes, $T_{c,-569}$; the transit center of the last follow-up photometry $T_{c,+123}$, the radial velocity semi-amplitude K, the light curve parameters $R_p/R_\star$, b^2 and $\zeta/R_\star$. Additionally, the out-of-transit magnitudes (both for the HATNet photometry and the two follow-up photometry), the zero-point of the radial velocity γ, and the EPD coefficients for the two follow-up photometry were also included in the fit. The EPD was performed up to the first order against the profile sharpness parameters (S, D, K), the hour angle and the airmass. In the case of the HATNet photometry, we incorporated an additional parameter, an instrumental blend factor whose inclusion was based on the experience that HATNet light curves tend to slightly underestimate the depth of the transits. To have a general purpose analysis, we extended the parameter set with the Lagrangian orbital elements $k = e\cos\omega$ and $h = e\sin\omega$, but based our assumption for circular orbits, these were fixed to be zero in the case of HAT-P-7b.

The XMMC analysis was performed in three ways. First, a full XMMC run was accomplished, involving all of the 23 parameters discussed below (6 physical parameters, 3 out-of-transit magnitudes, the radial velocity zero-point, the 2×5 EPD coefficients, the instrumental blend factor and the fixed Lagrangian orbital elements). Second, we have separated the 2×5 linear EPD coefficients from the merit function and run the Markov chains while minimizing the χ^2 accordingly in each step of the chain. Third, we derived the best fit parameters using the downhill simplex algorithm and during the XMMC run we kept the EPD coefficients to be fixed to their best fit values. All of these fits yielded a successful

Table 3.2: Stellar parameters for HAT-P-7. The values of effecitve temperature, metallicity and projected rotational velocity are based on purely spectroscopic data while the other ones are derived from the both the spectroscopy and the joint light curve and stellar evolution modelling.

Parameter	Value	Source	Parameter	Value	Source
T_{eff} (K)	6350 ± 80	SME	$\log g_\star$ (cgs)	$4.05^{+0.04}_{-0.06}$	Y^2+LC+SME
[Fe/H]	$+0.26 \pm 0.08$	SME	$L_\star$ ($L_\odot$)	$5.3^{+1.1}_{-0.6}$	Y^2+LC+SME
$v \sin i$ (km s^{-1})	3.8 ± 0.5	SME	M_V (mag)	2.91 ± 0.16	Y^2+LC+SME
$M_\star$ ($M_\odot$)	$1.49^{+0.06}_{-0.05}$	Y^2+LC+SME	Age (Gyr)	2.1 ± 1.0	Y^2+LC+SME
$R_\star$ ($R_\odot$)	$1.92^{+0.17}_{-0.11}$	Y^2+LC+SME	Distance (pc)	320^{+30}_{-20}	Y^2+LC+SME

convergence and all of the sanity checks mentioned in Sec. 2.12.16 were adequate, namely a) the *a posteriori* distribution centers of the adjusted parameters (median values) agreed well with the downhill simplex best fit values, b) the chain acceptance ratio was in agreement with the theoretical expectations, c) the correlation lengths for the parameter chains were sufficiently small, all of them were smaller than ~ 2.6, and d) the covariance estimations from the Fisher information matrix agreed well, within a factor of ~ 1.2, with the statistical covariances derived from the *a posteriori* distributions. See also Fig. 3.5, that shows the (slightly clarified and simplified) output of the lfit program related to this particular analysis. In all of the cases, we have used a Gaussian *a priori* distribution for the transitions, where the covariance matrix of this Gaussian were derived from the Fisher matrix evaluated at the downhill simplex best fit value. In Fig. 3.6 the distributions and some statistics for the 6 parameters related to the physical planetary (and orbital) parameters are displayed. The plots in Fig. 3.6 clearly show how the proper selection of the adjusted parameters can help to reduce the mutual correlations. The only significant correlation is between $\zeta/R_\star$ $T_{\text{c},+123}$. This correlation is resulted from the lack of a good quality complete follow-up photometry (due to its large scatter, the contribution of the second follow-up light curve is relatively smaller).

For the final set of the parameters we accepted the distribution that was derived using the third method mentioned above (i.e. when in the XMMC runs the 2×5 EPD parameters were fixed to their best fit values). The derived best fit parameters that are related to physical quantities were the following: $T_{\text{c},-569} = 2453153.09286 \pm 0.00105$ (BJD), $T_{\text{c},+123} = 2454678.76582 \pm 0.00137$ (BJD), $K = 213.4 \pm 1.9 \, \text{m s}^{-1}$, $p = R_{\text{p}}/R_\star = 0.7619 \pm 0.0009$, $b^2 = 0.206 \pm 0.103$ and $\zeta/R_\star = 13.45 \pm 0.22 \, \text{day}^{-1}$. Comparing to these values with the ones presented in Sec. 3.3.2, the improvements in the parameter uncertainties are quite conspicuous. Especially, the new, image subtraction based HATNet light curve has decreased the uncertainty in the first transit epoch of $T_{\text{c},-569}$ with its significantly better quality. In the further analysis, we incorporated these distributions in order to derive the final stellar, planetary and orbital parameters.

3.3.4 Stellar parameters

The results of the joint fit enable us to refine the parameters of the star. First, the iodine-free template spectrum from Keck was used for an initial determination of the atmospheric parameters. Spectral synthesis modeling was carried out using the SME software (Valenti & Piskunov, 1996), with wavelength ranges and atomic line data as described by Valenti & Fischer (2005). We obtained the following initial values: effective temperature $6350 \pm 80\,\text{K}$, surface gravity $\log g_\star = 4.06 \pm 0.10$ (cgs), iron abundance [Fe/H] $= +0.26 \pm 0.08$, and projected rotational velocity $v \sin i = 3.8 \pm 0.5\,\text{km s}^{-1}$. The rotational velocity is slightly smaller than the value given by the DS measurements. The temperature and surface gravity correspond to a slightly evolved F6 star. The uncertainties quoted here and in the remaining of this discussion are twice the statistical uncertainties for the values given by the SME analysis. This reflects our attempt, based on prior experience, to incorporate systematic errors (e.g. Noyes et al. (2008); see also Valenti & Fischer (2005)). Note that the previously discussed limb darkening coefficients, $\gamma_1^{(z)}$, $\gamma_2^{(z)}$, $\gamma_1^{(I)}$ and $\gamma_2^{(I)}$ have been taken from the tables of Claret (2004) by interpolation to the above-mentioned SME values for T_{eff}, $\log g_\star$, and [Fe/H].

As described by Sozzetti et al. (2007), $a/R_\star$ is a better luminosity indicator than the spectroscopic value of $\log g_\star$ since the variation of stellar surface gravity has a subtle effect on the line profiles. Therefore, we used the values of T_{eff} and [Fe/H] from the initial SME analysis, together with the distribution of $a/R_\star$ to estimate the stellar properties from comparison with the Yonsei-Yale (Y^2) stellar evolution models by Yi et al. (2001). Since a Monte-Carlo set for $a/R_\star$ values has been derived during the joint fit, we performed the stellar parameter determination as follows. For a selected value of $a/R_\star$, two Gaussian random values were drawn for T_{eff} and [Fe/H] with the mean and standard deviation as given by SME (with formal SME uncertainties doubled as indicated above).Using these three values, we searched the nearest isochrone and the corresponding mass by using the interpolator provided by Demarque et al. (2004). Repeating this procedure for values of $a/R_\star$, T_{eff}, [Fe/H], the set of the *a posteriori* distribution of the stellar parameters was obtained, including the mass, radius, age, luminosity and color (in multiple bands). The age determined in this way is 2.2 Gy with a statistical uncertainty of ± 0.3 Gy; however, the uncertainty in the theoretical isochrone ages is about 1.0 Gy. Since the corresponding value for the surface gravity of the star, $\log g_\star = 4.05^{+0.04}_{-0.06}$ (cgs), is well within 1-σ of the value determined by the SME analysis, we accept the values from the joint fit as the final stellar parameters. These parameters are summarized in Table 3.2.

We note that the Yonsei-Yale isochrones contain the absolute magnitudes and colors for different photometric bands from U up to M, providing an easy comparison of the estimated and the observed colors. Using these data, we determined the $V - I$ and $J - K$ colors of the

best fitted stellar model: $(V-I)_{YY} = 0.54\pm0.02$ and $(J-K)_{YY} = 0.27\pm0.02$. Since the colors for the infrared bands provided by Yi et al. (2001) and Demarque et al. (2004) are given in the ESO photometric standard system, for the comparison with catalog data, we converted the infrared color $(J-K)_{YY}$ to the 2MASS system $(J-K_S)$ using the transformations given by Carpenter (2001). The color of the best fit stellar model was $(J-K_S)_{YY} = 0.25\pm0.03$, which is in fairly good agreement with the actual 2MASS color of HAT-P-7: $(J-K_S) = 0.22\pm0.04$. We have also compared the $(V-I)_{YY}$ color of the best fit model to the catalog data, and found that although HAT-P-7 has a low galactic latitude, $b_{II} = 13°.8$, the model color agrees well with the observed TASS color of $(V-I)_{TASS} = 0.60 \pm 0.07$ (see Droege et al., 2006). Hence, the star is not affected by the interstellar reddening within the errors, since $E(V-I) \equiv (V-I)_{TASS} - (V-I)_{YY} = 0.06\pm0.07$. For estimating the distance of HAT-P-7, we used the absolute magnitude $M_V = 2.91 \pm 0.16$ (resulting from the isochrone analysis, see also Table 3.2) and the $V_{TASS} = 10.51 \pm 0.06$ observed magnitude. These two yield a distance modulus of $V_{TASS} - M_V = 7.51 \pm 0.28$, i.e. distance of $d = 320^{+30}_{-20}$ pc.

3.3.5 Planetary and orbital parameters

The determination of the stellar properties was followed by the characterization of the planet itself. Since Monte-Carlo distributions were derived for both the light curve and the stellar parameters, the final planetary and orbital data were also obtained by the statistical analysis of the *a posteriori* distribution of the appropriate combination of these two Monte-Carlo data sets. We found that the mass of the planet is $M_p = 1.800^{+0.063}_{-0.059} M_J$, the radius is $R_p = 1.421^{+0.144}_{-0.097} R_J$ and its density is $\rho_p = 0.78 \pm 0.16 \, \mathrm{g \, cm^{-3}}$. We note that in the case of binary systems with large mass and radius ratios (such as the one here) there is a strong correlation between M_p and R_p (see e.g. Beatty et al., 2007). This correlation is also exhibited here with $C(M_p, R_p) = 0.81$. The final planetary parameters are also summarized at the bottom of Table 3.3.

Due to the way we derived the period, i.e. $P = (T_{c,-29} - T_{c,-569})/540$, one can expect a large correlation between the epochs $T_{c,-29}$, $T_{c,-569}$ and the period itself. Indeed, $C(T_{c,-569}, P) = -0.783$ and $C(T_{c,-29}, P) = 0.704$, while the correlation between the two epochs is relatively small; $C(T_{c,-569}, T_{c,-29}) = -0.111$. It is easy to show that if the signs of the correlations between two epochs T_A and T_B (in our case $T_{c,-29}$ and $T_{c,-569}$) and the period are different, respectively, then there exists an optimal epoch E, which has the smallest error among all of the interpolated epochs. We note that E is such that it also exhibits the smallest correlation with the period. If $\sigma(T_A)$ and $\sigma(T_B)$ are the respective uncorrelated errors of the two epochs, then

$$E = \left[\frac{T_A \sigma(T_B)^2 + T_B \sigma(T_A)^2}{\sigma(T_B)^2 + \sigma(T_A)^2} \right] \tag{3.3}$$

Table 3.3: Orbital and planetary parameters for HAT-P-7. The parameters are derived from the joint modelling of the photometric, radial velocity and spectroscopic data.

Parameter	Value	Parameter	Value
P (days)	2.2047298 ± 0.0000024	K (m s^{-1})	213.2 ± 1.9
E (BJD $-$ 2,400,000)	$53,785.8503 \pm 0.0008$	γ (km s^{-1})	-37.0 ± 1.5
T_{14} (days)[a]	0.1625 ± 0.0029	e	0 (adopted)
$T_{12} = T_{34}$ (days)[a]	0.0141 ± 0.0020	M_p ($M_{\rm J}$)	$1.800^{+0.063}_{-0.059}$
$a/R_\star$	$4.25^{+0.24}_{-0.28}$	R_p ($R_{\rm J}$)	$1.421^{+0.144}_{-0.097}$
$R_p/R_\star$	0.0761 ± 0.0009	$C(M_p, R_p)$	0.81
$b \equiv a \cos i/R_\star$	$0.44^{+0.10}_{-0.15}$	ρ_p (g cm^{-3})	0.78 ± 0.16
i (deg)	$84°.1^{+2.2}_{-2.0}$	a (AU)	0.0379 ± 0.0004
Transit duration (days)	0.1461 ± 0.0016	$\log g_p$ (cgs)	3.34 ± 0.07
(γ_1, γ_2) [b]	$(0.1195, 0.3595)$	$T_{\rm eq}$ (K)	2175^{+85}_{-60}

[a] T_{14}: total transit duration, time between first to last contact; $T_{12} = T_{34}$: ingress/egress time, time between first and second, or third and fourth contact.

where square brackets denote the time of the transit event nearest to the time instance t. In the case of HAT-P-7b, $T_{\rm A} \equiv T_{\rm c,-569}$ and $T_{\rm B} \equiv T_{\rm c,-29}$, the corresponding epoch is the event $N_{\rm tr} = -280$ at $E \equiv T_{\rm c,-280} = 2,453,785.8503 \pm 0.0008$ (BJD). The final ephemeris and planetary parameters are summarized in Table 3.3.

Chapter 4

Follow-up observations

Now we shift our attention to another system, one of the eccentric transiting planetary systems of HAT-P-2b. At the time of its discovery, HAT-P-2b was the longest period and most massive transiting extrasolar planet (TEP), and the only one on an eccentric orbit (Bakos et al., 2007b). In the following, other TEPs have also been discovered with significant orbital eccentricity, and long period: GJ 436b (Gillon et al., 2007), HD 17156b (Barbieri et al., 2007) and XO-3b (Johns-Krull et al., 2008). (See, e.g. `http://exoplanet.eu` for an up-to-date database for transiting extrasolar planets.)

Planet HAT-P-2b was detected as a transiting object during the campaign of the HATNet telescopes (Bakos et al., 2002, 2004), and Wise HAT telescope (WHAT Shporer et al., 2006). The HATNet telescopes and the WHAT telescope gathered $\sim 26,000$ individual photometric measurements. The planetary transit was followed up by the FLWO 1.2m telescope, utilizing the KeplerCam detector. The planetary properties have been confirmed by radial velocity measurements and bisector analysis of the spectral line profiles. The latter has shown no bisector variations, excluding the possibilities of a hierarchical triplet or a blended eclipsing binary.

Recently, the spin-orbit alignment of the HAT-P-2(b) system was measured by Winn et al. (2007) and Loeillet et al. (2008). Both studies reported an alignment consistent with zero within an uncertainty of $\sim 10°$. These results are exceptionally interesting since short period planets are thought to be formed at much larger distances from their parent star and migrated inward while the orbital eccentricity is damped yielding an almost circular orbit (D'Angelo, Lubow & Bate, 2006). Physical mechanisms such as Kozai interaction between the transiting planet and an unknown massive companion on an inclined orbit could result tight eccentric orbits (Fabrycky & Tremaine, 2007; Takeda, Kita & Rasio, 2008). However, in such a scenario, the spin-orbit alignment can be expected to be significantly larger than the measured. For instance, in the case of XO-3b (Hébrard et al., 2008), the reported alignment is $\lambda = 70° \pm 15°$. In multiple planetary systems, planet-planet scattering can also yield

eccentric orbits (see e.g. Ford & Holman, 2007).

The physical properties of the host star HAT-P-2 have been controversial since different methods for stellar characterization resulted stellar radii between $\sim 1.4\,R_\odot$ and $\sim 1.8\,R_\odot$. Moreover, the actual distance of the system also had large systematic errors, since the reported Hipparcos distance seemed to be significantly larger than what could be expected from the absolute luminosity (coming from the stellar evolution modelling).

In this chapter new photometric and spectroscopic observations of the planetary system HAT-P-2(b) are presented, and I demonstrate how the photometry package can be used in the case of a follow-up observation. The new photometric measurements significantly improve the light curve parameters, therefore some of the stellar parameters are more accurately constrained. In addition, radial velocity measurements based on spectroscopic observations have resulted significantly smaller uncertainties, which, due to the orbital eccentricity, also affect the results of the stellar evolution modelling. In Sec. 4.1, we summarize our photometric observations of this system, while in Sec. 4.2 we describe briefly the issues related to the radial velocity data points. The details of a new formalism used in the characterization of the radial velocities is discussed in Sec. 4.3 and the steps of the complete analysis are described in Sec. 4.4. We summarize our results in Sec. 4.5.

4.1 Photometric observations and reductions

In the present analysis we utilize photometric data obtained by the HATNet telescopes (published in Bakos et al., 2007b) and by the KeplerCam detector mounted on the FLWO 1.2m telescope. The photometry of HATNet have already been presented in Bakos et al. (2007b). These HATNet data are plotted on Fig. 4.1, superimposed with our new best-fit model (see Sec. 4.4 for details on light curve modelling). We observed the planetary transit six times, on 2007 March 18, 2007 April 21, 2007 May 08, 2007 June 22, 2008 March 24 and 2008 May 25, yielding 4 nearly complete and 2 partial transit light curves. One of these follow-up light curves (2007 April 21) has already been published in the discovery paper. All of our high precision follow-up photometry data are plotted on Fig. 4.2, along with our best-fit transit light curve model (see also Sec. 4.4).

The frames taken by the KeplerCam detector have been calibrated and reduced in the following similar fashion for all of the observations for the six nights. Prior to the real calibration, all pixels which are saturated (or blooming) have been marked (`fiign`, see Sec. 2.12.6), forcing them to be omitted from the upcoming photometry. During the calibration of the frames we have used standard bias, dark and sky-flat corrections.

Following the calibration, the detection of stars and the derivation of the astrometrical solution was done in two steps. First, an initial astrometrical transformation was derived
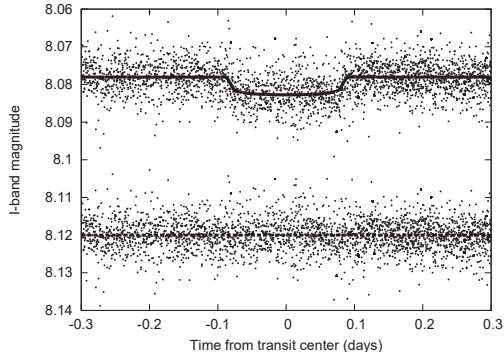
Figure 4.1: The folded HATNet light curve of HAT-P-2 (published in Bakos et al., 2007b), showing the points only nearby the transit. The upper panel is superimposed with our best-fit model and the lower panel shows the fit residual. See text for further details.

using the ~ 50 brightest and non-saturated stars (whose parameters were derived by the program `fistar`, see Sec. 2.12.8) from each frame, and by using the 2MASS catalogue (Skrutskie, 2006) as a reference. The transformation itself has been obtained by the program `grmatch` (Sec. 2.12.10), with a second-order polynomial fit. Since the astrometrical data found in the 2MASS catalogue was obtained by the same kind of telescope, one could expect significantly better astrometrical data from the FLWO 1.2 m telescope due to the numerous individual frames taken at better spatial resolution. Indeed, an internal catalog which was derived from the detected stellar centroids by registering them to the same reference system has shown an internal precision ~ 0.005 arc sec for the brighter stars while the 2MASS catalog reports an uncertainty that is larger by an order of magnitude: nearly ~ 0.06 arc sec. Therefore, in the second step of the astrometry, we used this new catalog to derive the individual astrometrical solutions for each frame, still using a second-order polynomial fit. We note here that this method also corrects for the systematic errors in the photometry yielded by the proper motion of the stars.

Using the above astrometrical solutions, we performed aperture photometry (with the program `fiphot`, Sec. 2.12.13) on fix centroids, employing a set of five apertures between 7.5 and 17.5 pixels in radius. The results of the aperture photometry were then transformed to the same instrumental magnitude system using a correction to the spatial variations and the differential extinction (the former depends on the celestial coordinates while the latter depends on the intrinsic colors of the stars). Both corrections were linear in the pixel coordinates and linear in the colors. Experience shows that significant correlations can occur between the instrumental magnitudes and some of the external parameters of the light curves (such as the FWHM of the stars, subpixel positions). Although one should de-trend against
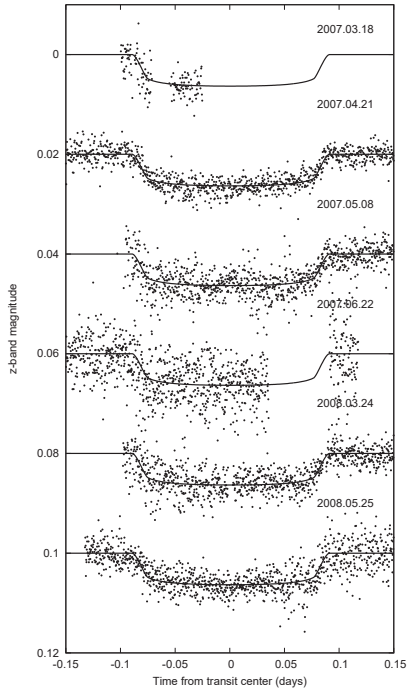
Figure 4.2: Follow-up light curves of HAT-P-2. The light curves were acquired on 2007 March 18, 2007 April 21, 2007 May 08, 2007 June 22, 2008 March 24 and 2008 May 25, while the respective transit sequence numbers were $N_{tr} = -6$, 0, +3, +11, +60 and +71. All of these light curves are superimposed with our best-fit model. See text for further details.

these correlations using purely out-of-transit data (both before ingress and after egress), we have carried out such an external parameter decorrelation (EPD) simultaneously with the light curve modelling (Sec. 4.4) due to the lack of out-of-transit data in several cases. After the simultaneous light curve modelling and de-trending, we chose the aperture for each night that yielded the smallest residual. In all of the cases this "best aperture" was neither the smallest nor the largest one from the set, confirming our assumptions for selecting a good aperture series. We note here that since all of the stars on the frames were well isolated, such choice of different radii of the apertures does not result in any systematics, because stars are not blended by any of these apertures. In addition, due to the high flux of HAT-P-2 and the comparison stars, the frames were slightly extrafocal (in order to avoid saturation). This resulted different FWHM per night for the stars and therefore the optimal apertures yielding the highest signal-to-noise ratio also have different radii for each night.

Table 4.1: Comprehensive list of relative radial velocity measurements for HAT-P-2. The Keck measurements marked with an asterix and the Lick measurements are published in Bakos et al. (2007b). The OHP/SOPHIE data are taken from Loeillet et al. (2008).

BJD − 2M4	RV (m/s)	σ_{RV}(m/s)	Source	BJD − 2M4	RV (m/s)	σ_{RV}(m/s)	Source
53981.77748	12.0	7.3	Keck[*]	54547.11569	554.6	7.4	Keck
53982.87168	−288.3	7.9	Keck[*]	54549.05046	784.8	9.2	Keck
53983.81485	569.0	7.3	Keck[*]	54602.91654	296.3	7.0	Keck
54023.69150	727.3	7.8	Keck[*]	54603.93210	688.0	5.9	Keck
54186.99824	721.3	7.7	Keck[*]	54168.96790	−152.7	42.1	Lick[a]
54187.10415	711.0	6.7	Keck[*]	54169.95190	542.4	41.3	Lick[a]
54187.15987	738.1	6.8	Keck[*]	54170.86190	556.8	42.6	Lick[a]
54188.01687	783.6	7.1	Keck[*]	54171.03650	719.1	49.6	Lick[a]
54188.15961	801.8	6.7	Keck[*]	54218.80810	−1165.2	88.3	Lick[a]
54189.01037	671.0	6.7	Keck[*]	54218.98560	−1492.6	90.8	Lick[a]
54189.08890	656.7	6.8	Keck[*]	54219.93730	−28.2	43.9	Lick[a]
54189.15771	640.2	6.9	Keck[*]	54219.96000	−14.8	43.9	Lick[a]
54216.95938	747.7	8.1	Keck	54220.96410	451.6	38.4	Lick[a]
54279.87688	402.0	8.3	Keck	54220.99340	590.7	37.1	Lick[a]
54285.82384	168.3	5.7	Keck	54227.50160	−19401.4	8.8	OHP[b]
54294.87869	756.8	6.5	Keck	54227.60000	−19408.2	6.5	OHP[b]
54304.86497	615.5	6.2	Keck	54228.58420	−19558.1	18.8	OHP[b]
54305.87010	764.2	6.3	Keck	54229.59930	−20187.4	16.1	OHP[b]
54306.86520	761.4	7.6	Keck	54230.44750	−21224.9	14.1	OHP[b]
54307.91236	479.1	6.5	Keck	54230.60290	−20853.6	14.8	OHP[b]
54335.81260	574.7	6.8	Keck	54231.59870	−19531.1	12.1	OHP[b]
54546.09817	−670.9	10.1	Keck	54236.51900	−20220.7	5.6	OHP[b]

4.2 Radial velocity observations

In the discovery paper of HAT-P-2b (Bakos et al., 2007b), 13 individual radial velocity measurements were reported that were utilizing the HIRES instrument (Vogt et al., 1994) on the Keck I telescope, on Mauna Kea, Hawaii, plus 10 measurements from the Hamilton echelle spectrograph at the Lick Observatory (Vogt, 1987). In the last year, we have acquired 14 additional radial velocity measurements using the HIRES instrument on Keck. In the analysis, we have also used the online radial velocity data obtained by the OHP/SOPHIE spectrograph at out-of-transit (i.e. omitting the measurements for the Rossiter-McLaughlin effect), published by Loeillet et al. (2008). With these additional 8 observations, we have $27 + 10 + 8 = 45$ high precision RV data points at hand for a refined analysis.

In Table 4.1 we collected all (previously published and our newly obtained) radial velocity measurements. In Fig. 4.4 we show the RV data, overplotted with our best-fit model solution (for details of the fit, see Sec. 4.4).

4.3 An analytical formalism for Kepler's problem

In this section we present a set of analytic relations (based on a few smooth functions defined in a closed form) that provides a straightforward solution of Kepler's problem, and consequently, time series of RV data and RV model functions. Due to the analytic property, the partial derivatives can also be obtained directly and therefore can be utilized in various fitting and data analysis methods, including the Fisher analysis of uncertainties and correlations. The functions presented here are nearly as simple to manage as trigonometric functions. This section has three major parts. In Sec. 4.3.1, the basics of the mathematical formalism are presented, including the rules for calculating partial derivatives. In Sec. 4.3.2, the solution of the spatial problem is shown, supplemented with the inverse problem, still using infinitely differentiable functions. This part also discusses how transits constrain the phase of the radial velocity curve. And finally, in Sec. 4.3.3, we show how the presented formalism can be implemented in practice, in the framework of the `lfit` program and involving some of the built-in functions.

4.3.1 Mathematical formalism

The solution for the time evolution of Kepler's problem can be derived in the standard way as given in various textbooks (see, e.g., Murray & Dermott, 1999). The restricted two body problem itself is an integrable ordinary differential equation. In the planar case, three independent integrals of motion exist and one variable with uniform monotonicity (i.e. which is an affine function of time). The integrals are related to the well known orbital elements, that are used to characterize the orbit. These are the semimajor axis a, the eccentricity e and the longitude of pericenter[1] ϖ. The fourth quantity is the mean anomaly $M = nt$, where $n = \sqrt{\mu/a^3} = 2\pi/P$, the mean motion, which is zero at pericenter passage[2]. The solution to Kepler's problem can be given in terms of the mean anomaly M as defined as

$$E - e \sin E = M, \qquad (4.1)$$

where E is the eccentric anomaly. The spatial coordinates are

$$\xi = \xi_0 \cos \varpi - \eta_0 \sin \varpi, \qquad (4.2)$$

$$\eta = \xi_0 \sin \varpi + \eta_0 \cos \varpi, \qquad (4.3)$$

where

$$\xi_0 = a(\cos E - e), \qquad (4.4)$$

$$\eta_0 = a\sqrt{1 - e^2} \sin E; \qquad (4.5)$$

---

[1] In two dimensions, the argument of pericenter is always equal to the longitude of pericenter, i.e. $\varpi \equiv \omega$

[2] The mass parameter of Kepler's problem is denoted by $\mu \equiv \mathcal{G}(m_1 + m_2)$, where m_1 and m_2 are the masses of the two orbiting bodies and $\mathcal{G}$ is the Newtonian gravitational constant.

see also Murray & Dermott (1999), Sect. 2.4 for the derivation of these equations. Since for circular orbits the longitude of pericenter and pericenter passage cannot be defined, and for nearly circular orbits, these can only be badly constrained; in these cases it is useful to define a new variable, the mean longitude as $\lambda = M + \varpi$ to use instead of M. Since ϖ is an integral of the motion, $\dot{\lambda} = \dot{M} = n$. Therefore for circular orbits $\varpi \equiv 0$ and equations (4.4)-(4.5) should be replaced by

$$\xi_0 = a \cos \lambda, \tag{4.6}$$

$$\eta_0 = a \sin \lambda. \tag{4.7}$$

To obtain an analytical solution to the problem, i.e. which is infinitely differentiable with respect to all of the orbital elements and the mean longitude, first let us define the Lagrangian orbital elements $k = e \cos \varpi$ and $h = e \sin \varpi$. Substituting equations (4.4)-(4.5) into equations (4.2)-(4.3) gives

$$\begin{pmatrix} \xi \\ \eta \end{pmatrix} = a \left[\begin{pmatrix} c \\ s \end{pmatrix} + \frac{e \sin E}{2 - \ell} \begin{pmatrix} +h \\ -k \end{pmatrix} - \begin{pmatrix} k \\ h \end{pmatrix} \right], \tag{4.8}$$

where $c = \cos(\lambda + e \sin E)$, $s = \sin(\lambda + e \sin E)$ and $\ell = 1 - \sqrt{1 - e^2}$, the oblateness of the orbit. The derivation of the above equation is straightforward, one should only keep in mind that $E + \varpi = \lambda + e \sin E$. In the first part of this section we prove that the quantities

$$\mathrm{p}(\lambda, k, h) = \begin{cases} 0 & \text{if } k = 0 \text{ and } h = 0 \\ e \sin E & \text{otherwise} \end{cases} \tag{4.9}$$

and

$$\mathrm{q}(\lambda, k, h) = \begin{cases} 0 & \text{if } k = 0 \text{ and } h = 0 \\ e \cos E & \text{otherwise} \end{cases} \tag{4.10}$$

are analytic – infinitely differentiable – functions of λ, k and h for all real values of λ and for all $k^2 + h^2 = e^2 < 1$. In the following parts, we utilize the partial derivatives of these analytic functions to obtain the orbital velocities, and we also derive some other useful relations. In this section we only deal with planar orbits, the three dimensional case is discussed in the next section.

Partial derivatives and the analytic property

A real function is analytic when all of its partial derivatives exist, the partial derivatives are continuous functions and only depend on other analytic functions. It is proven in Pál (2009) that the partial derivatives of $q = \mathrm{q}(\lambda, k, h)$ and $p = \mathrm{p}(\lambda, k, h)$ are the following for $(k, h) \neq (0, 0)$:

$$\frac{\partial q}{\partial \lambda} = \frac{-p}{1 - q}, \tag{4.11}$$

$$\frac{\partial q}{\partial k} = \frac{c - k}{1 - q} = \frac{\cos(\lambda + p) - k}{1 - q}, \tag{4.12}$$

$$\frac{\partial q}{\partial h} = \frac{s - h}{1 - q} = \frac{\sin(\lambda + p) - h}{1 - q} \tag{4.13}$$

and

$$\frac{\partial p}{\partial \lambda} = \frac{q}{1 - q}, \tag{4.14}$$

$$\frac{\partial p}{\partial k} = \frac{+s}{1 - q} = \frac{+\sin(\lambda + p)}{1 - q}, \tag{4.15}$$

$$\frac{\partial p}{\partial h} = \frac{-c}{1 - q} = \frac{-\cos(\lambda + p)}{1 - q}. \tag{4.16}$$

Since for all $k^2 + h^2 < 1$, $q < 1$ and therefore $1 - q > 0$, all of the above functions are continuous on their domains. Since the $\sin(\cdot)$ and $\cos(\cdot)$ functions are analytic, therefore one can conclude that the functions $q(\cdot, \cdot, \cdot)$ and $p(\cdot, \cdot, \cdot)$ are also analytic.

Substituting the definition of $p = p(\lambda, k, h)$ into equation (4.8), one can write

$$\begin{pmatrix} \xi \\ \eta \end{pmatrix} = a \left[\begin{pmatrix} \cos(\lambda + p) \\ \sin(\lambda + p) \end{pmatrix} + \frac{p}{2 - \ell} \begin{pmatrix} +h \\ -k \end{pmatrix} - \begin{pmatrix} k \\ h \end{pmatrix} \right], \tag{4.17}$$

while the radial distance of the orbiting body from the center is $\sqrt{\xi^2 + \eta^2} = r = a(1 - q)$. For small eccentricities in equation (4.17) the third term (k, h) is negligible compared to the first term $(\cos, \sin)$ while the second term $(h, -k)p/(2 - \ell)$ is negligible compared to the third term. Therefore for $e \ll 1$, p is proportional to the phase offset in the polar angle of the orbiting particle (as defined from the geometric center of the orbit) and q is proportional to the distance offset relative to a circular orbit; both caused by the non-zero orbital eccentricity.

Since equation (4.17) is a combination of purely analytic functions, the solution of Kepler's problem is analytic with respect to the orbital elements a, (k, h), and to the mean longitude λ in the domain $a > 0$ and $k^2 + h^2 < 1$. We note here that this formalism omits the parabolic or hyperbolic solutions. The formalism based on the Stumpff functions (see Stiefel & Scheifele, 1971) provides a continuous set of formulae for the elliptic, parabolic, and hyperbolic orbits but this parametrization is still singular in the $e \to 0$ limit.

Orbital velocities

Assuming a non-perturbed orbit, i.e. when $(\dot{k}, \dot{h}) = 0$, and $\dot{a} = 0$ and when the mean motion $n = \dot{\lambda}$ is constant, the orbital velocities can be directly obtained by calculating the partial derivative of equation (4.17) with respect to λ and applying the chain rule since

$$\frac{\partial}{\partial t} \begin{pmatrix} \xi \\ \eta \end{pmatrix} \equiv \begin{pmatrix} \dot{\xi} \\ \dot{\eta} \end{pmatrix} = \left[\frac{\partial}{\partial \lambda} \begin{pmatrix} \xi \\ \eta \end{pmatrix} \right] \frac{\partial \lambda}{\partial t} = n \frac{\partial}{\partial \lambda} \begin{pmatrix} \xi \\ \eta \end{pmatrix}. \tag{4.18}$$

Substituting the partial derivative equation (4.14) into the expansion of $\partial\xi/\partial\lambda$ and $\partial\eta/\partial\lambda$ one gets

$$\begin{pmatrix} \dot{\xi} \\ \dot{\eta} \end{pmatrix} = \frac{an}{1-q} \left[\begin{pmatrix} -\sin(\lambda+p) \\ +\cos(\lambda+p) \end{pmatrix} + \frac{q}{2-\ell} \begin{pmatrix} +h \\ -k \end{pmatrix} \right]. \tag{4.19}$$

Note that equation (4.19) is also a combination of purely analytic functions, the components of the orbital velocity are analytic with respect to the orbital elements a, (k, h), and to the mean longitude λ.

It is also evident that the time derivative of equation (4.19) is

$$\begin{pmatrix} \ddot{\xi} \\ \ddot{\eta} \end{pmatrix} = \frac{-an^2}{(1-q)^3} \left[\begin{pmatrix} \cos(\lambda+p) \\ \sin(\lambda+p) \end{pmatrix} + \right. \tag{4.20}$$
$$\left. + \frac{p}{2-\ell} \begin{pmatrix} +h \\ -k \end{pmatrix} - \begin{pmatrix} k \\ h \end{pmatrix} \right].$$

Obviously, equation (4.20) can be written as

$$\begin{pmatrix} \ddot{\xi} \\ \ddot{\eta} \end{pmatrix} = -\frac{n^2}{(1-q)^3} \begin{pmatrix} \xi \\ \eta \end{pmatrix}, \tag{4.21}$$

which is equivalent to the equations of motion since $\mu = n^2 a^3$ and $\sqrt{\xi^2 + \eta^2} = r = a(1-q)$.

4.3.2 Additional constraints given by the transits

In the follow-up observations of planets discovered by transits in photometric data series, the detection of variations in the RV signal is one of the most relevant steps, either to rule out transits of late-type dwarf stars, and/or blends, or to characterize the mass of the planet and the orbital parameters. Since transit timing constrains the epoch and orbital period much more precisely than radial velocity alone, these two can be assumed to be fixed in the analysis of the RV data. However, this constraint also includes an additional feature. The mean longitude has to be shifted to the transits since it is $\pi/2$ only for circular orbits at the time of the transit. It can be shown that the mean longitude at the time instance of the transit is

$$\lambda_{\mathrm{tr}} = \arg\left(k + \frac{kh}{2-\ell}, 1 + h - \frac{k^2}{2-\ell} \right) - \frac{k(1-\ell)}{h}, \tag{4.22}$$

therefore the mean longitude at the orbital phase φ becomes $\lambda = \lambda_{\mathrm{tr}} + 2\pi\varphi$. Thus, the observed radial velocity signal is proportional to the $\dot{\eta}$ component of the velocity vector, namely

$$\mathrm{RV} = \gamma + K_0 v, \tag{4.23}$$
$$v = \dot{\eta}(\lambda_{\mathrm{tr}} + 2\pi\varphi, k, h), \tag{4.24}$$

where γ is the mean barycentric velocity and K_0 is related to the semi-amplitude K as $K_0 = K\sqrt{1-e^2}$. Consequently, the partial derivatives of the $v = \dot{\eta}$ RV component, $v =$

```
lfit    -x "eoc(l,k,h)=cos(1+eop(l,k,h))" \
        -x "eos(l,k,h)=sin(1+eop(l,k,h))" \
        -x "J(k,h)=sqrt(1-k*k-h*h)" \
        -x "lamtranxy(x,y,k,h)=arg(k+x+h*(k*y-h*x)/(1+J(k,h)),h+y-k*(k*y-h*x)/(1+J(k,h)))-
            (k*y-h*x)*J(k,h)/(1+k*x+h*y)" \
        -x "lamtran(l0,k,h)=lamtranxy(cos(l0),sin(l0),k,h)" \
        -x "prx0(l,k,h)=(+eoc(l,k,h)+h*eop(l,k,h)/(1+J(k,h))-k)" \
        -x "pry0(l,k,h)=(+eos(l,k,h)-k*eop(l,k,h)/(1+J(k,h))-h)" \
        -x "rvx0(l,k,h)=(-eos(l,k,h)+h*eoq(l,k,h)/(1+J(k,h)))/(1-eoq(l,k,h))" \
        -x "rvy0(l,k,h)=(+eoc(l,k,h)-k*eoq(l,k,h)/(1+J(k,h)))/(1-eoq(l,k,h))" \
        -x "prx1(l,k,h)=prx0(1+lamtranxy(0,1,k,h),k,h)" \
        -x "pry1(l,k,h)=pry0(1+lamtranxy(0,1,k,h),k,h)" \
        -x "rvx1(l,k,h)=rvx0(1+lamtranxy(0,1,k,h),k,h)" \
        -x "rvy1(l,k,h)=rvy0(1+lamtranxy(0,1,k,h),k,h)" \
        -x "rvbase(l,k,h)=rvy1(l,k,h)" \
        ...
```

Figure 4.3: Macro definitions for lfit, implementing some functions related to radial velocity analysis. All of the above functions are based on the eccentric offset functions eop(.,.,.) and eoq(.,.,.) as defined by equations (4.9) and (4.10).

$\dot\eta(\lambda_\mathrm{tr} + 2\pi\varphi, k, h)$ with respect to the orbital elements k and h are

$$\frac{\partial v}{\partial k} = \frac{\partial \dot\eta}{\partial k} + \frac{\partial \dot\eta}{\partial \lambda}\frac{\partial \lambda_\mathrm{tr}}{\partial k}, \tag{4.25}$$

$$\frac{\partial v}{\partial h} = \frac{\partial \dot\eta}{\partial h} + \frac{\partial \dot\eta}{\partial \lambda}\frac{\partial \lambda_\mathrm{tr}}{\partial h}. \tag{4.26}$$

A radial velocity curve of a star, caused by the perturbation of a single companion can be parametrized by six quantities: the semi-amplitude of RV variations, K, the zero point, G, the Lagrangian orbital elements, (k, h), the epoch, T_0 (or equivalently the phase at an arbitrary fixed time instant) and the period P. In the cases of transiting planets, the later two are known since the photometric observations of the transits constrain both quantities with exceeding precision (relative to the precision attainable purely by the RV data). Therefore, one has to fit only four quantities, i.e. $\mathbf{a} = (K, G, k, h)$.

4.3.3 Practical implementation

The eccentric offset functions $p(\lambda, k, h)$ and $q(\lambda, k, h)$ are implemented in the program lfit (see also Sec. 2.12.16). This program does not provide further functionality related to the radial velocity analysis, however, the macro definition capabilities of the program can be involved in order to define some more useful functions which then can be directly applied in real problems. The shell script pieces shown in Fig. 4.3 demonstrate how equations (4.22) and (4.24) are implemented in practice. The parametric derivatives of these functions, such as equation (4.25) or (4.26) are then derived automatically by lfit, using the partial derivatives of the base functions $p(\lambda, k, h)$ and $q(\lambda, k, h)$ as well as the chain rule.

4.4 Analysis of the HAT-P-2 planetary system

In this section we briefly describe the analysis of the available photometric and radial velocity data of HAT-P-2 in order to determine the planetary parameters as accurately as possible. The modelling was done in three major steps in an iterative way. The first step was the modelling of the light curve and the radial velocity data series. Second, this was followed by the determination of the stellar parameters. In the last step, by combining the light curve parameters with the stellar properties, we obtained the physical parameters (mass, radius) of the planet.

To model transit light curves taken in optical or near-infrared photometric passbands, we include the effect of the stellar limb darkening. We have used the formalism of Mandel & Agol (2002) to model the flux decrease during transits under the assumption of quadratic limb darkening law. Since the limb darkening coefficients are the function of the stellar atmospheric parameters (such as effective temperature $T_{\rm eff}$, surface gravity $\log g_\star$ and metallicity), the whole light curve analysis should be preceded by the initial derivation of these parameters. These parameters were obtained by collaborators, using the iodine-free template spectrum obtained by the HIRES instrument on Keck I and employing the Spectroscopy Made Easy software package (Valenti & Piskunov, 1996), supported by the atomic line database of Valenti & Fischer (2005). This analysis yields the $T_{\rm eff}$, $\log g_\star$, [Fe/H] and the projected rotational velocity $v \sin i$. The result of the SME analysis when all of these values have been adjusted simultaneously were $\log g_\star = 4.22 \pm 0.14$ (CGS), $T_{\rm eff} = 6290 \pm 110$ K, [Fe/H] $= 0.12 \pm 0.08$ and $v \sin i = 20.8 \pm 0.2$ km s^{-1}.

The limb darkening coefficients are then derived for z' and I photometric bands by interpolation, using the tables provided by Claret (2000) and Claret (2004). The initial values for the coefficients were $\gamma_1^{(z)} = 0.1430$, $\gamma_2^{(z)} = 0.3615$, $\gamma_1^{(I)} = 0.1765$, and $\gamma_2^{(I)} = 0.3688$. After the first iteration, with the knowledge of the stellar parameters, the SME analysis is repeated by fixing the surface gravity to the value yielded by the stellar evolution modelling. This can be done in a straightforward way: the normalized semimajor axis $a/R_\star$ can be obtained from the transit light curve model parameters, the orbital eccentricity and the argument of pericenter. As it was pointed out by Sozzetti et al. (2007), the ratio $a/R_\star$ is a more effective luminosity indicator than the stellar surface gravity, since the stellar density is related to

$$\rho_\star \propto (a/R_\star)^3. \tag{4.27}$$

Since HAT-P-2b is a quite massive planet, i.e. $M_{\rm p}/M_\star \sim 0.01$, relation (4.27) requires a significant correction, which also depends on observable quantities (see Pál et al., 2008b, for more details). In our case, this correction is not negligible since $M_{\rm p}/M_\star$ is comparable to the typical relative uncertainties in the light curve parameters.
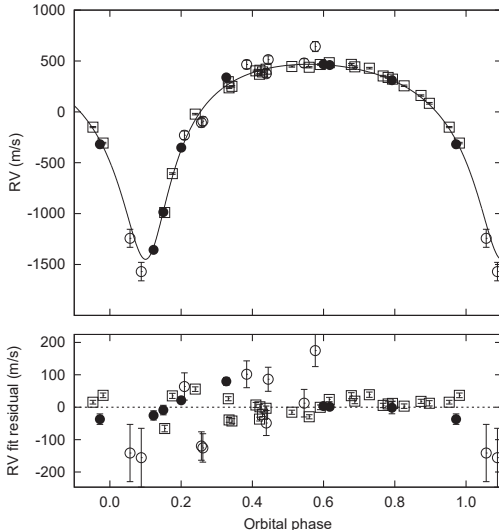
Figure 4.4: Radial velocity measurements for HAT-P-2 folded with the best-fit orbital period. Filled dots represent the OHP/SOPHIE data, open circles show the Lick/Hamilton, while the open boxes mark the Keck/HIRES observations. In the upper panel, all of these three RV data sets are shifted to zero mean barycentric velocity. The RV data are superimposed with our best-fit model. The lower panel shows the residuals from the best-fit. Note the different vertical scales on the two panels. The transit occurs at zero orbital phase. See text for further details.

4.4.1 Light curve and radial velocity parameters

The first major step of the analysis is the determination of the light curve and radial velocity parameters. We performed a joint fit by adjusting the light curve and radial velocity parameters simultaneously as described below.

The parameters can be classified into three major groups. The light curve parameters that are related to the *physical* properties of the planetary system are the transit epoch E, the period P, the fractional planetary radius $p \equiv R_p/R_\star$, the impact parameter b, and the normalized semimajor axis $a/R_\star$. The physical radial velocity parameters are the RV semi-amplitude K, the orbital eccentricity e and the argument of pericenter ω. In the third group there are parameters that are not related to the physical properties of the system, but are rather instrumentation specific ones. These are the out-of-transit instrumental magnitudes of the follow-up (and HATNet) light curves, and the RV zero-points γ_{Keck}, γ_{Lick} and γ_{OHP} of the three individual data sets[3].

---

[3]Since in the reduction of the Loeillet et al. (2008) data a synthetic stellar spectrum was used as a reference, γ_{OHP} is the physical barycentric radial velocity of the system. In the reductions of the Keck and Lick data, we used one of the spectra as a template, therefore the zero-points of these two are arbitrary, lack

To minimize the correlation between the adjusted parameters, we use a slightly different parameter set. Instead of adjusting the epoch and period, we fitted the first and last available transit center time, T_{-148} and T_{+71}. Here indices note the transit event number: the $N_{\mathrm{tr}} \equiv 0$ event was defined as the first complete follow-up light curve taken on 2007 April 21, the first available transit observation from the HATNet data was the event $N_{\mathrm{tr}} = -148$ and the last follow-up was observed on 2008 May 25, was event $N_{\mathrm{tr}} \equiv +71$. Note that assuming equidistant transit cadences, all of the transit centers available in the HATNet and follow-up photometry are constrained by these two transit instances (see Bakos et al., 2007c; Pál et al., 2008a). Similarly, instead of the eccentricity e and argument of pericenter ω, we have adjusted the Lagrangian orbital elements $k \equiv e \cos \omega$ and $h \equiv e \sin \omega$. These elements show no correlation in practice, moreover, the radial velocity curve is an analytic function of these even for $e \to 0$ cases (although in the case of HAT-P-2b this is irrelevant because e is non-zero). As it is known in the literature (Winn et al., 2007b; Pál, 2008), the impact parameter b and $a/R_\star$ are also strongly correlated, especially for small $p \equiv R_{\mathrm{p}}/R_\star$ values. Therefore, as it was suggested by Bakos et al. (2007c), we chose the parameters $\zeta/R_\star$ and b^2 for fitting instead of $a/R_\star$ and b, where for eccentric orbits $\zeta/R_\star$ is related to $a/R_\star$ as

$$\frac{\zeta}{R_\star} = \left(\frac{a}{R_\star} \right) \frac{2\pi}{P} \frac{1}{\sqrt{1-b^2}} \frac{\sqrt{1-e^2}}{1+h}. \tag{4.28}$$

The quantity $\zeta/R_\star$ is related to the transit duration as $T_{\mathrm{dur}} = 2(\zeta/R_\star)^{-1}$, if the duration is defined between the time instants when the center of the planet crosses the limb of the star inwards and outwards.

4.4.2 Effects of the orbital eccentricity

Let us denote the projected radial distance between the center of the planet and the center of the star (normalized by $R_\star$) by d. As it was shown in Pál (2008), d can be parametrized in a second order approximation as

$$d^2 = (1 - b^2) \left(\frac{\zeta}{R_\star} \right)^2 (\Delta t)^2 + b^2, \tag{4.29}$$

where Δt is the time between the actual observation time and the intrinsic transit center. The *intrinsic transit* center is defined when the planet reaches its maximal tangential velocity during the transit. Although the tangential velocity cannot be measured directly, the intrinsic transit center is determined by purely the radial velocity data, without any knowledge of the transit geometry[4]. For eccentric orbits the impact parameter b is related to the

---

any real physical interpretation.

[4]In other words, predictions can only be made for the intrinsic transit center in cases where the planet was discovered by a radial velocity survey and initially we have no further constraint for the geometry of the system.

orbital inclination i as

$$b = \left(\frac{a}{R_\star}\right) \cos i \frac{1 - e^2}{1 + h}. \tag{4.30}$$

In order to have a better description of the transit light curve, we used a higher order expansion in the $d(\Delta t)$ function (Eq. 4.29). For circular orbits, such an expansion is straightforward. To derive the expansion for elliptic orbits, we employed the method of Lie-integration which gives the solution of any ordinary differential equation (here, the equations for the two-body problem) in a recursive series for the Taylor expansion with respect to the independent variable (here, the time). It can be shown involving the Taylor expansion of the orbital motion that the normalized projected distance d up to fourth order is:

$$
\begin{aligned}
d^2 &= b^2 \left[1 - 2R\varphi - (Q - R^2)\varphi^2 - \frac{1}{3}QR\varphi^3 \right] + \\
&\quad \left(\frac{\zeta}{R_\star}\right)^2 (1 - b^2)\Delta t^2 \left[1 - \frac{1}{3}Q\varphi^2 + \frac{1}{2}QR\varphi^3 \right],
\end{aligned}
\tag{4.31}
$$

where

$$Q = \left(\frac{1 + h}{1 - e^2}\right)^3, \tag{4.32}$$

and

$$R = \frac{1 + h}{(1 - e^2)^{3/2}}k. \tag{4.33}$$

Here $n = 2\pi/P$ is the mean motion, and φ is defined as $\varphi = n\Delta t$. For circular orbits, $Q = 1$ and $R = 0$, and for small eccentricities ($e \ll 1$), $Q \approx 1 + 3h$ and $R \approx k$. The leading order correction term in φ, $-2b^2 R\varphi$, is related to the time lag between the photometric and intrinsic transit centers. The *photometric transit center* is defined halfway between the instants when the center of the planet crosses the limb of the star inward and outward. It is easy to show by solving the equation $d(\varphi) = 1$, yielding two solutions (φ_{I} and φ_{E}), that this phase lag is:

$$
\begin{aligned}
\Delta\varphi &= \frac{\varphi_{\mathrm{I}} + \varphi_{\mathrm{E}}}{2} = \tag{4.34} \\
&= -\frac{b^2 R}{\left(\frac{\zeta}{R_\star}\frac{1}{n}\right)^2 (1 - b^2) - (Q - R^2)b^2} \approx \tag{4.35} \\
&\approx -\left(\frac{a}{R_\star}\right)^{-2} \frac{b^2 k}{(1 + h)\sqrt{1 - e^2}}, \tag{4.36}
\end{aligned}
$$

which can result in a time lag of several minutes.

In equation (4.31), the third order terms in φ describe the asymmetry between the slopes of the ingress and egress parts of the light curve. For some some other aspects of light curve asymmetries see Loeb (2005) and Barnes (2007). In the cases when no assumptions are known for the orbital eccentricity, we cannot treat the parameters R and Q as independent

since the intrinsic transit center and R have an exceptionally high correlation. However, if we assume a simpler model function, with only third order terms in φ with fitted coefficients present, i.e.

$$
\begin{aligned}
d^2 &= b^2 \left[1 - \varphi^2 - \frac{1}{3} C \varphi^3 \right] + \\
&\quad \left(\frac{\zeta}{R_\star} \right)^2 (1 - b^2) \Delta t^2 \left[1 - \frac{1}{3} \varphi^2 + \frac{1}{2} C \varphi^3 \right],
\end{aligned}
\tag{4.37}
$$

yields a non-zero value for the C coefficient for asymmetric light curves. In the case of HAT-P-2b, the derived values for Q and R are $Q = 2.23 \pm 0.10$ and $R = -0.789 \pm 0.021$ (derived from the values of k and h, see Sec. 4.4.3), thus the coefficient for the third order term in φ is $QR = -1.75 \pm 0.13$. Using equation (4.37), for an "ideal" light curve (with similar parameters of k, h, $\zeta/R_\star$ and b^2 as for HAT-P-2b), the best fit value for C is $C = -2.23$, which is close to the value of $QR \approx -1.75$. The difference between the best fit value of C and the fiducial value of QR is because in equation (4.37) the coefficients for the first and second order terms were fixed to be 0 and 1, respectively. Although this asymmetry can be measured directly (without leading to any degeneracy between the fit parameters), in practice we need extreme photometric precision to obtain a significant detection for a non-zero C parameter: assuming a photometric time series for a single transit of HAT-P-2b with 5 sec cadence where each individual measurement has a photometric error of 0.01 mmag(!), the uncertainty in C is ± 0.47, equivalent to a 5-σ detection of the light curve asymmetry. This detection would be hard for ground-based instrumentation (i.e. for a 1-σ detection one should achieve a photometric precision of 0.05 mmag at the same cadence). Space missions like *Kepler* (Borucki et al., 2007) will be able to detect orbital eccentricity relying only on photometry of primary transits.

4.4.3 Joint fit

As it was discussed before, in order to achieve a self-consistent fit, we performed a simultaneous fit on all of the light curve and radial velocity data. We have involved equation (4.31), to model the light curves, where the parameters Q and R were derived from the actual values of k and h, using equations equation (4.32) and equation (4.33). To find the best-fit values for the parameters we used the downhill simplex algorithm (see Press et al., 1992) and we used the method of refitting to synthetic data sets to get an *a posteriori* distribution for the adjusted values. The final results of the fit were $T_{-148} = 2453379.10281 \pm 0.00141$, $T_{+71} = 2454612.83271 \pm 0.00075$, $K = 958.9 \pm 13.9 \, \mathrm{m \, s^{-1}}$, $k = -0.5119 \pm 0.0040$, $h = -0.0543 \pm 0.0098$, $R_\mathrm{p}/R_\star \equiv p = 0.0724 \pm 0.0010$, $b^2 = 0.125 \pm 0.073$, $\zeta/R_\star = 12.090 \pm 0.046 \, \mathrm{day^{-1}}$, $\gamma_\mathrm{Keck} = 318.4 \pm 6.6 \, \mathrm{m \, s^{-1}}$, $\gamma_\mathrm{Lick} = 77.0 \pm 30.4 \, \mathrm{m \, s^{-1}}$, $\gamma_\mathrm{OHP} = -19868.9 \pm 9.8 \, \mathrm{m \, s^{-1}}$. The uncertainties of the out-of-transit magnitudes were be-

Table 4.2: Stellar parameters for HAT-P-2. The values of effecitve temperature, metallicity and projected rotational velocity are based on purely spectroscopic data (SME) while the other ones are derived from the both the spectroscopy (SME) and the joint modelling (LC+Y^2).

Parameter	Value	Source	Parameter	Value	Source
T_{eff} (K)	6290 ± 60	SME[a]	$\log g_\star$ (cgs)	4.158 ± 0.031	Y^2+LC+SME
[Fe/H]	$+0.14 \pm 0.08$	SME	$L_\star$ ($L_\odot$)	$3.6^{+0.5}_{-0.3}$	Y^2+LC+SME
$v \sin i$ (km s^{-1})	20.8 ± 0.3	SME	M_V (mag)	3.36 ± 0.12	Y^2+LC+SME
$M_\star$ ($M_\odot$)	1.34 ± 0.04	Y^2+LC+SME[a]	Age (Gyr)	2.7 ± 0.5	Y^2+LC+SME
$R_\star$ ($R_\odot$)	$1.60^{+0.09}_{-0.07}$	Y^2+LC+SME	Distance (pc)	118 ± 8	Y^2+LC+SME

tween $(6 \ldots 21) \times 10^{-5}$ mag for the follow-up light curves and 16×10^{-5} mag for the HATNet data. The fit resulted a normalized χ^2 value of 0.995. As it is described in the following subsection, the resulted distribution has been used then as an input for the stellar evolution modelling.

4.4.4 Stellar parameters

The second step of the analysis was the derivation of the physical stellar parameters. Following the complete Monte-Carlo way of parameter estimation, as it was described by Pál et al. (2008a), we calculated the distribution of the stellar density, derived from the $a/R_\star$ values. To be more precise, the density of the star is

$$\rho_\star = \rho_0 - \frac{\Sigma_0}{R_\star},$$
(4.38)

where both ρ_0 and Σ_0 are directly related to observable quantities, namely

$$\rho_0 = \frac{3\pi}{GP^2} \left(\frac{a}{R_\star} \right)^3,$$
(4.39)

$$\Sigma_0 = \frac{3K\sqrt{1-e^2}}{2PG \sin i} \left(\frac{a}{R_\star} \right)^2.$$
(4.40)

In equation (4.38), the only unknown quantity is the radius of the star, which can be derived using a stellar evolution model, and it depends on a luminosity indicator (that is, in practice, the surface gravity or the density of the star), a color indicator (which is the T_{eff} effective surface temperature, given by the SME analysis) and the stellar composition (here [Fe/H]). Therefore, one can write

$$R_\star = R_\star(\rho_\star, T_{\mathrm{eff}}, [\mathrm{Fe/H}]).$$
(4.41)

Since both T_{eff} and [Fe/H] are known from stellar atmospheric analysis, equation (4.38) and equation (4.41) have two unknowns, and thus this set of equations can be solved iteratively. Note that in order to solve equation (4.41), supposing its parameters are known in advance,

one has to use a certain stellar evolutionary model. Such models are available in tabulated form, therefore the solution of the equation requires the inversion of the interpolating function on the tabulated data. Thus, equation (4.41) is only a symbolical notation for the algorithm which provides the solution. Moreover, if the star is evolved, the isochrones and/or evolutionary tracks for the stellar models intersect themselves, resulting an ambiguous solution (i.e. it is not a "function" any more). For HAT-P-2, however, the solution of equation (4.41) is definite since the host star is a main sequence star. To obtain the physical parameters (e.g. the stellar radius), we used the stellar evolutionary models of Yi et al. (2001), by interpolating the values of $\rho_\star$, T_{eff} and [Fe/H] using the interpolator provided by Demarque et al. (2004).

The procedure described above has been applied to all of the parameters in the input set, where the values of ρ_0 have been derived from the values of $a/R_\star$ and the orbital period P using equation (4.39), while the values for T_{eff} and [Fe/H] have been drawn from Gaussian random variables with the mean and standard deviation of the first SME results ($T_{\mathrm{eff}} = 6290\pm110$ K and [Fe/H] $= 0.12\pm0.08$). This step resulted the *a posteriori* distribution of the physical stellar parameters, including the surface gravity. The value and uncertainty for the latter was $\log g_\star = 4.16\pm0.04$ (CGS), which is slightly smaller than the value provided by the SME analysis. To reduce the uncertainties in T_{eff} and [Fe/H], we repeated the SME modelling by fixing the value of $\log g_\star$ to the above. This second SME run resulted $T_{\mathrm{eff}} = 6290 \pm 60$ K and [Fe/H] $= 0.14 \pm 0.08$. Following, we updated the values for the limb darkening parameters ($\gamma_1^{(z)} = 0.1419$, $\gamma_2^{(z)} = 0.3634$, $\gamma_1^{(I)} = 0.1752$, and $\gamma_2^{(I)} = 0.3707$), and repeated the simultaneous light curve and radial velocity fit. The results of this fit were then used to repeat the stellar evolution modelling, which yielded among other parameters $\log g_\star = 4.158 \pm 0.031$ (CGS). Since the value of $\log g_\star$ did not change significantly, we accepted these stellar parameter values as final ones. The stellar parameters are summarized in Table 4.2 and the light curve and radial velocity parameters are listed in the top two blocks of Table 4.3.

4.4.5 Planetary parameters

In the previous two steps of the analysis, we determined the light curve, radial velocity and stellar parameters. In order to get the planetary parameters, we combined the two Monte-Carlo data sets that yield their *a posteriori* distribution in a consistent way. For example, the mass of the planet is calculated using

$$M_{\mathrm{p}} = \frac{2\pi}{P} \frac{K\sqrt{1-e^2}}{G\sin i} \left(\frac{a}{R_\star}\right)^2 R_\star^2, \qquad (4.42)$$

where the values for the period P, RV semi-amplitude K, eccentricity e, inclination i, and normalized semimajor axis $a/R_\star$ were taken from the results of the light curve and RV fit

while the values for $R_\star$ were taken from the respective points of the stellar parameter distribution. From the distribution of the planetary parameters, we obtained the mean values and uncertainties. We derived $M_p = 8.84^{+0.22}_{-0.29} M_{Jup}$ for the planetary mass, $R_p = 1.123^{+0.071}_{-0.054} R_{Jup}$ for the radius while the correlation between these parameters were $C(M_p, R_p) = 0.68$. The planetary parameters are summarized in the lower block of Table 4.3.

Due to the eccentric orbit and the lack of the knowledge of the heat redistribution of the incoming stellar flux, the surface temperature of the planet can be constrained with difficulties. Assuming complete heat redistribution, the surface temperature can be estimated by time averaging the incoming flux which varies as $1/r^2 = a^{-2}(1 - e \cos E)^{-2}$ due to the orbital eccentricity. The time average of $1/r^2$ is

$$\left\langle \frac{1}{r^2} \right\rangle = \frac{1}{T} \int_0^T \frac{dt}{r^2(t)} = \frac{1}{2\pi} \int_0^{2\pi} \frac{dM}{r^2(M)}, \qquad (4.43)$$

where M is the mean anomaly of the planet. Since $r = a(1 - e \cos E)$ and $dM = (1 - e \cos E)dE$, where E is the eccentric anomaly, the above integral can be calculated analytically and the result is

$$\left\langle \frac{1}{r^2} \right\rangle = \frac{1}{a^2\sqrt{1 - e^2}}. \qquad (4.44)$$

Using this time averaged weight for the incoming flux, we derived $T_p = 1525^{+40}_{-30}$ K. However, the planet surface temperature would be ~ 2975 K on the dayside during periastron and assuming no heat redistribution, while the equilibrium temperature would be only ~ 1190 K if the planet was always at that of apastron. Thus, we conclude that the surface temperature can vary by a factor of ~ 3, depending on the actual atmospheric dynamics.

4.4.6 Photometric parameters and the distance of the system

The stellar evolution modelling (see Sec. 4.4.4) also yields the absolute magnitudes and colors for the models for various photometric passbands. We compared the obtained colors and absolute magnitudes with other observations. First, the $V - I$ color of the modelled star was compared with the observations. The TASS catalogue (Droege et al., 2006) has magnitudes for this star, $V_{TASS} = 8.71 \pm 0.04$ and $I_{TASS} = 8.16 \pm 0.05$, i.e. the observed color of the star is $(V - I)_{TASS} = 0.55 \pm 0.06$. The stellar evolution modelling resulted a color of $(V - I)_{YY} = 0.552 \pm 0.016$, which is in perfect agreement with the observations. The absolute magnitude of the star in V band is $M_V = 3.36 \pm 0.12$, also given by the stellar evolution models. This therefore yields a distance modulus of $V_{TASS} - M_V = 5.35 \pm 0.13$, which is equivalent to a distance of 117 ± 7 pc, assuming no interstellar reddening. This distance value for the star is placed right between the distance values found in the two different available Hipparcos reductions of Perryman et al. (1997) and van Leeuwen (2007a,b): Perryman et al.

Table 4.3: Spectroscopic and light curve solutions for HAT-P-2, and inferred planet parameters, derived from the joint modelling of photometric, spectroscopic and radial velocity data.

Parameter	Value	Parameter	Value
P (days)	5.6334697 ± 0.0000074	$a/R_\star$	$9.21^{+0.37}_{-0.40}$
E (HJD$-2{,}400{,}000$)	$54{,}342.42616 \pm 0.00064$	b	$0.354^{+0.087}_{-0.156}$
T_{14} (days)[a]	0.1790 ± 0.0013	i (deg)	$87^\circ.2^{+1.2\circ}_{-0.9}$
$T_{12} = T_{34}$ (days)[a]	0.0136 ± 0.0012	$M_{\rm p}$ ($M_{\rm Jup}$)	$8.84^{+0.22}_{-0.29}$
$R_{\rm p}/R_\star$	0.0724 ± 0.0010	$R_{\rm p}$ ($R_{\rm Jup}$)	$1.123^{+0.071}_{-0.054}$
K (m s^{-1})	958.9 ± 13.9	$C(M_{\rm p}, R_{\rm p})$	0.68
$k \equiv e\cos\omega$	-0.5119 ± 0.0040	$\rho_{\rm p}$ (g cm^{-3})	$7.63^{+1.14}_{-1.09}$
$h \equiv e\sin\omega$	-0.0543 ± 0.0098	a (AU)	0.0686 ± 0.0007
e	0.5148 ± 0.0038	$\log g_{\rm p}$ (cgs)	4.23 ± 0.04
ω	$186.1^\circ \pm 1.1^\circ$	$T_{\rm eff}$ (K)	1525^{+40}_{-30} (see [b])

[a] T_{14}: total transit duration, time between first to last contact; $T_{12} = T_{34}$: ingress/egress time, time between first and second, or third and fourth contact.

[b] This effective temperature assumes uniform heat redistribution while the irradiance is averaged on the orbital revolution. See text for further details about the issue of the planetary surface temperature.
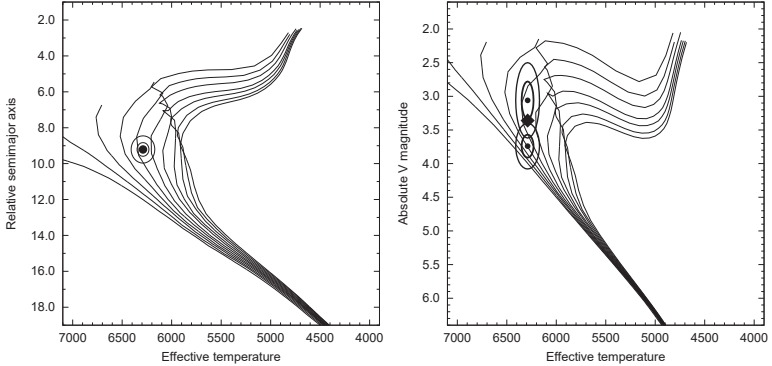

Figure 4.5: Stellar evolutionary isochrones from the Yonsei-Yale models, showing the isochrones for [Fe/H] = 0.14 stars, between 0.5 and 5.5 Gyrs (with a cadence of 0.5 Gyrs). The stellar color is indicated by the effective temperature, while the left panel shows the luminosity using the absolute V magnitude M_V and the right panel uses the ratio $a/R_\star$ as a luminosity indicator. In the left panel, the isochrones are overplotted by the 1-σ and 2-σ confidence ellipsoids, defined by the effective temperature, and the absolute magnitude estimations from the TASS catalogue and the two Hipparcos reductions (older: upper ellipse, recent: lower ellipse). The diamond indicates the M_V magnitude derived from our best fit stellar evolution models. On the right plot, the confidence ellipsoid of the effective temperature and $a/R_\star$ is shown.

(1997) reports a parallax of 7.39 ± 0.88 mas, equivalent to a distance of 135 ± 18 pc while van Leeuwen (2007a,b) states a parallax of 10.14 ± 0.73 mas, equivalent to a distance of 99 ± 7 pc. In the two panels of Fig. 4.5, stellar evolutionary isochrones are shown for the metallicity of HAT-P-2, superimposed by the effective temperature and various luminosity estimations

based on both the above discussion (relying only on various Hipparcos distances and TASS apparent magnitudes) and the constraints yielded by the stellar evolution modelling. The 2MASS magnitude of the star in J band is $J_{\mathrm{2MASS}} = 7.796 \pm 0.027$ while the stellar evolution models yielded an absolute magnitude of $M_J = 2.465 \pm 0.110$. Thus, the distance modulus here is $J_{\mathrm{2MASS}} - M_J = 5.33 \pm 0.11$, equivalent to a distance of $116 \pm 6 \, \mathrm{pc}$, confirming the distance derived from the photometry taken from the TASS catalogue.

4.5 Discussion

We presented refined planetary, stellar and orbital parameters for the HAT-P-2(b) transiting extrasolar planetary system. Our improved analysis was based on numerous radial velocity data points, including both new measurements and data taken from the literature. We have also carried out high precision follow-up photometry. The refined parameters have uncertainties that are smaller by a factor of ~ 2 in the planetary parameters and a factor of $\sim 3-4$ in the orbital parameters than the previously reported values of Bakos et al. (2007b). We note that the density of the planet turned out to be significantly smaller that the value by Bakos et al. (2007b), namely $\rho_{\mathrm{p}} = 7.6 \pm 1.1 \, \mathrm{g \, cm^{-3}}$, moreover, the uncertainty reported by Bakos et al. (2007b) was significantly larger. In our analysis we did not rely on the distance of the system, i.e. we did not use the absolute magnitude as a luminosity indicator. Instead, our stellar evolution modelling was based on the density of the star, an other luminosity indicator related to precise light curve and RV parameters. We have compared the estimated distance of the system (which was derived from the absolute magnitudes, known from the stellar modelling) with the Hipparcos distances. We found that our newly estimated distance falls between the two values available from the different reductions of Hipparcos raw data.

The improved orbital eccentricity and argument of pericenter allow us to estimate the time of the possible secondary transits. We found that secondary transits occur at the orbital phase of $\phi_{\mathrm{sec}} = 0.1886 \pm 0.0020$, i.e. 1 day 1 hour and 30 minutes ($\pm$ 16 minutes) after primary transit events.

The zero insolation planetary isochrones of Baraffe et al. (2003) give an expected radius of $R_{\mathrm{p,Baraffe03}} = 1.02 \pm 0.02 \, R_{\mathrm{Jup}}$, that is slightly smaller than the measured radius of $1.12^{+0.07}_{-0.05} \, R_{\mathrm{Jup}}$. The work of Fortney et al. (2007) takes into account not only the evolutionary age and the total mass of the planet but the incident stellar flux and the mass of the planet's core. By scaling the semimajor axis of HAT-P-2b to one that yields the same incident flux from a solar-type star on a circular orbit, taking into account both the luminosity of the star and the correction for the orbital eccentricity given by equation (4.44), we obtained $a' = 0.033 \pm 0.003 \, \mathrm{AU}$. Using this scaled semimajor axis, the interpolation based on the tables provided by Fortney et al. (2007) yields radii between

$R_{\text{p,Fortney,0}} = 1.142 \pm 0.003\,R_{\text{Jup}}$ (core-less planets) and $R_{\text{p,Fortney,100}} = 1.111 \pm 0.003\,R_{\text{Jup}}$ (core-dominated planets, with a core of $M_{\text{p,core}} = 100\,M_{\oplus}$). Although these values agree nicely with our value of $R_{\text{p}} = 1.123^{+0.071}_{-0.054}\,R_{\text{Jup}}$, the relatively large uncertainty of R_{p} excludes any further conclusion for the size of the planet's core. Recent models of Baraffe, Chabrier & Barman (2008) also give the radius of the planet as the function of evolutionary age, metal enrichment and an optional insolation for equivalent to scaled semimajor axis of $a' = 0.045$ AU. Using this latter insolation, their models yield $R_{\text{p,Baraffe08,0.02}} = 1.055 \pm 0.006\,R_{\text{Jup}}$ (for metal poor, $Z = 0.02$ planets) and $R_{\text{p,Baraffe08,0.10}} = 1.008 \pm 0.006\,R_{\text{Jup}}$ (for more metal rich, $Z = 0.10$ planets). These values are slightly smaller than the actual radius of HAT-P-2b, however, the actual insolation of HAT-P-2b is roughly two times larger than the insolation implied by $a' = 0.045$ AU. Since the respective planetary radii of Baraffe, Chabrier & Barman (2008) for zero insolation give $R^{(0)}_{\text{p,Baraffe08,0.02}} = 1.009 \pm 0.006\,R_{\text{Jup}}$ and $R^{(0)}_{\text{p,Baraffe08,0.10}} = 0.975 \pm 0.006\,R_{\text{Jup}}$ for the respective cases of $Z = 0.02$ and $Z = 0.10$ metal enrichment, an extrapolation for a two times larger insolation would put the expected planetary radius in the range of $\sim 1.10\,R_{\text{Jup}}$. This is consistent with the models of Fortney et al. (2007) as well as with the measurements. However, as discussed earlier in the case of Fortney et al. (2007) models, the uncertainty in R_{p} does not let us properly constrain the metal enrichment.

HAT-P-2b will remain an interesting target, as a member of an emerging heavy-mass population. Further photometric measurements will refine the light curve parameters and therefore more precise stellar parameters can also be obtained. This will yield smaller uncertainties in the physical planetary radius, thus some parameters of the planetary evolution models, such as the metal enrichment can be obtained more explicitly. Moreover, observations of secondary eclipses will reveal the planetary atmosphere temperature which now is poorly constrained. Since the secondary eclipse occurs shortly after periastron passage, the temperature and therefore the contrast might be high enough to detect the occultation with a good signal-to-noise ratio.

Chapter 5

Summary

Transiting extrasolar planets are the only group among the extrasolar planets whose basic physical parameters, such as mass and radius can be determined without any ambiguity. Therefore, these planets provide a great opportunity to determine other properties, such as the characteristics of the planetary interior or their atmosphere. Recently, wide-field photometric surveys became the most prominent observation techniques for detecting transiting planets and these surveys yielded several dozens of discoveries. Since such wide-field surveys yield massive amount of data which cannot be efficiently and consistently processed by the available existing software solutions, I started developing a new package in order to overcome the related problems. The development of this package has been related to the Hungarian-made Automated Telescope Network (HATNet) project, one of the most successful initiatives searching for transiting extrasolar planets.

The aims of my work were both implementing the algorithms related to the photometric reduction in a form of a standalone software package, as well as applying these programs in the analysis of the HATNet data. Additionally, the photometric reduction is intended to work on data obtained by other facilities, typically 1m-class telescopes (such as the 48" telescope at Fred Lawrence Whipple Observatory or the Schmidt telescope at the Piszkéstető Mountain Station).

Of course, both the confirmation of planetary candidates and the characterization of known objects require other types of technologies such as spectroscopy, radial velocity measurements and stellar evolution modelling. In order to perform a consistent determination of the planetary, orbital and stellar parameters of transiting exoplanetary systems, my work also focused on to include these additional types of measurements and methods in the data analysis.

In this PhD thesis I presented a new software package intended to perform photometric data reduction on massive amount of astronomical images. Existing software solutions do not provide a consistent framework for the reduction of images acquired by wide-field and

undersampled instrumentation. During the development of the related algorithms and the implementation, I focused on the issues related to these problems in order to have a homogeneous reduction environment, ranging from the calibration of frames to the final light curve generation and analysis. This new package has been successfully applied in processing the images of the HATNet and led to the discovery and confirmation of almost a dozen of transiting extrasolar planets.

Acknowledgments

First of all, I would like to thank my parents and family for their immutable support during the years of my studies and in my whole life.

I would like to thank Gáspár Bakos for inviting to the project and for the possibility to be a member of people working in the field of transiting extrasolar planets. I also thank my supervisor, Bálint Érdi for the opportunity to be a PhD student at the Eötvös University and for his help in the proofreading. I am grateful to the hospitality of the Harvard-Smithsonian Center for Astrophysics, where this work has been partially carried out. I would like to thank Gspr, his wife Krisztina Meiszel and and other friends, István Cziegler, Gábor Fűrész, Bence Kocsis, Mária Pető and Dávid Végh for their continuous and great help related to the life in Cambridge and around Boston.

I thank Brigitta Sipőcz for her comments, ideas for improvements and bug reports related to the data analysis programs. I also thank collaborators at the Harvard-Smithsonian CfA and the Konkoly Observatory, Joel Hartman, Gábor Kovács, Géza Kovács, Robert Noyes and Guillermo Torres for their help.

I would like to say thanks to my friends Balázs Dianiska, Ágnes Kóspál, András László and Judit Szulágyi for their encouragement and for their valuable comments on this dissertation. Like so, I thank Eric Agol, Daniel Fabrycky, Bence Kocsis and Joshua Winn for their help during the preparation of various articles related to this field of science. I also thank Edward Miller, one of my former roommates for his help on the earlier versions of the draft.

Last but not least, I would like to thank Péter Ábrahám, Ferenc Horvai and Csaba Kiss for their present support and the opportunity to continue the related research in the Konkoly Observatory.

Bibliography

Agol, E., Steffen, J., Sari, R., & Clarkson, W., 2005, MNRAS, 359, 567

Aigrain, S., Hodgkin, S., Irwin, J., Hebb, L., Irwin, M., Favata, F., Moraux, E., Pont, F., 2007, MNRAS, 375, 29

Akerlof, C., et al.2000, AJ, 119, 1901

Alard, C. & Lupton, R. H. 1998, ApJ, 503, 325

Alard, C. 2000, A&AS, 144, 363

Alonso, R., et al. 2004, ApJL, 613, L153

Alonso, R., et al. 2008, A&A, 482, 21

Bailes, M., Lyne, A. G. & Shemar, S. L. 1991, Nature, 352, L311

Bakos, G. Á. et al., 2002, PASP, 114, 974

Bakos, G. Á. et al., 2004, PASP, 116, 266

Bakos, G. Á. 2004, PhD Thesis

Bakos, G. Á. et al., 2007, ApJ, 656, 552

Bakos, G. Á. et al., 2007, ApJ, 670, 826

Bakos, G. Á. et al., 2007, ApJ, 671, L173

Bakos, G. Á. et al., 2008, ApJ, accepted (arXiv:0809.4295)

Bakos, G. Á., Torres, G., Pál, A. et al. 2009, ApJ, submitted (arXiv:0901.0282)

Baraffe, I. et al. 2003, A&A, 402, 701

Baraffe, I., Chabrier, G. and Barman, T. 2008, A&A, 482, 315

Barbieri, M. et al. 2007, A&A, 476, L13

Barge, P. et al., 2008, A&A, 482, 17

Barnes, J. W., 2007, PASP, 119, 986

Baluev, R. V., 2008, MNRAS, 389, 1375

Beatty, T. G. et al., 2007, ApJ, 663, 573

Bertin, E. & Arnouts, S. 1996, A&AS, 117, 393

Borucki, W. J. et al., 2007, ASP Conf. Ser., 366, 309

Bouchy, F. et al., 2005, A&A, 444, 15

Bramich, D. M. et al., 2005, MNRAS, 359, 1096

Bramich, D. M. 2008, MNRAS, 386, 77

Brown, T. M., Charbonneau, D., Gilliland, R. L., Noyes, R. W. & Burrows, A., 2001, ApJ, 552, 699

Brown, T. M., 2003, ApJL, 593, L125

Burke, Christopher J., Gaudi, B. Scott, DePoy, D. L., Pogge, Richard W., Pinsonneault, Marc H., 2004, AJ, 127, 2382

Burrows, A., Hubeny, I., Budaj, J., & Hubbard, W. B. 2007, ApJ, 661, 502

Butler, R. P., Marcy, G. W., Williams, E., McCarthy, C., Dosanjh, P. & Vogt, S. S., 1996, PASP, 108, 500

Butler, R. P., Marcy, G. W., Fischer, D. A., Brown, T. M., Contos, A. R., Korzennik, S. G., Nisenson, P., Noyes, R. W. 1999, ApJ, 526, 916

Butler, R. P., Marcy, G. W. 1996, ApJL, 464, 153

Butler, R. P. et al., 2004, ApJ, 617, 580

Butler, R. P., Wright, J., Marcy, G., Fischer, D., Vogt, S., Tinney, C., Jones, H., Carter, B., Johnson, J, McCarthy & C. Penny, A., 2006, ApJ, 646, 505

Calabretta, M. R., Greisen, E. W. 2002, A&A, 395, 1077

Cameron, A. C. et al. 2006, MNRAS, 373, 799

Cameron, A. C. et al. 2007, MNRAS, 375, 951

Carpenter, J. 2001, AJ, 121, 2851

Carter, J. A., Yee, J. C., Eastman, J., Gaudi, B. S. & Winn, J. N. 2008, ApJ, 689, 499

Charbonneau, D., Brown, T. M., Latham, D. W. & Major, M., 2000, ApJ, 529, 45

Charbonneau, D., Brown, T. M., Burrows, A. & Laughlin, G. 2006, in Conference Proceedings of Protostars and Planets V, B. Reipurth, D. Jewitt, and K. Keil (eds.), University of Arizona Press, Tucson, 701

Claret, A. 2000, A&A, 363, 1081

Claret, A. 2004, A&A, 428, 1001

Demarque, P., Woo, J.-H., Kim, Y.-C. & Yi, S. K. 2004, ApJ, 155, 667

D'Angelo, G., Lubow, S. H. & Bate, M. R., 2006, ApJ, 652, 1698

Droege, T. F., Richmond, M. W., & Sallman, M., 2006, PASP, 118, 1666

Fabrycky, D. & Tremaine, S. 2007, ApJ, 669, 1298

Fabrycky, D., 2008, ApJ, 677, L117

Finn, L. S., 1992, Phys. Rev. D, 46, 5236

Fischer, D. A., Marcy, G. W., Butler, R. P., Laughlin, G., Vogt, S. S. 2002, ApJ, 564, 1028

Fischer, D. A. et al., 2007, ApJ, 669, 1336

Ford, E., 2004, AJ, 129, 1706

Ford, E. B. & Holman, M. J., 2007, ApJ, 664, 51

Ford, E. 2008, AJ, 135, 1008

Ford, E. B., Rasio, F. A. 2007, ApJ, 686, 621

Fortney, J. J., Marley, M. S. & Barnes, W. 2007, ApJ, 659, 1661

Fressin, F., Guillot, T., Morello, V., & Pont, F. 2007, A&A, 475, 729

Gaudi, B. S. 2005, ApJ, 628, L73

Gillon, M. et al. 2007, A&A, 472, L13

Gionis, A.: Computational Geometry: Nearest Neighbor Problem (lecture notes)[1]

Groth, E. J. 1986, AJ, 91, 1244

---

[1]http://theory.stanford.edu/~nmishra/CS361-2002/lecture12-scribe.pdf

Guillot, T., Santos, N. C., Pont, F., Iro, N., Melo, C., & Ribas, I. 2006, A&A, 453, L21

Hansen, B., & Barman, T. 2007, ApJ, 671, 861

Hartman, J. D., Bakos, G. Á., Stanek, K. Z., Noyes, R. W. 2004, AJ, 128, 1761

Hartman, J. D., Gaudi, B. S., Holman, M. J., McLeod, B. A., Stanek, K. Z., Barranco, J. A., Pinsonneault, M. H., Meibom, S., Kalirai, J. S., 2008, ApJ, 675, 1233

Hébrard, G., et al. 2008, A&A, 488, 763

Henry, G. W., Marcy, G. W., Butler, R. P. & Vogt, S. S. 2000, ApJ, 529, 41

Holman, M. J. et al. 2007, ApJ, 664, 1185

Howell, S. B. 1989, PASP, 101, 616

Hut, P., 1981, A&A, 99, 126

Irwin, Jonathan, Aigrain, Suzanne, Hodgkin, Simon, Irwin, Mike, Bouvier, Jerome, Clarke, Cathie, Hebb, Leslie, Moraux, Estelle, 2006, MNRAS, 370, 954

Israel, H., Hessman, F. V. & Schuh, S. 2007, AN, 328, 16

Johns-Krull, C. M. et al., 2008, ApJ, 677, 657

Jordán, A. & Bakos, G. Á., 2008, ApJ, 685, 543

Joye, W. A. & Mandel, E., 2003, Astronomical Data Analysis Software and Systems XII, ASP Conference Series, Vol. 295, 489 (eds. H. E. Payne, R. I. Jedrzejewski, and R. N. Hook)

Kim, D.-W., Protopapas, P. Alcock, Ch., Byun, Y.-I. & Bianco, F 2008, MNRAS, accepted (arXiv:0812.1010)

Kochanski, G. P., Tyson, J. A. & Fischer, P 1996, AJ, 111, 1444

Konacki, M., Torres, G., Jha, S., Sasselov, D. D., 2003, Nature, 421, 507

Kovács, G., Zucker, S., & Mazeh, 2002, A&A, 391, 369

Kovács, G., Bakos, G. Á., & Noyes, R. W., 2005, MNRAS, 356, 557

Kovács, G. et al., 2007, ApJ, 670, L41

Landolt, A. U. 1992, AJ, 104, 340

Latham, D. W. 1992, in IAU Coll. 135, Complementary Approaches to Double and Multiple Star Research, ASP Conf. Ser. 32, eds. H. A. McAlister & W. I. Hartkopf (San Francisco: ASP), 110

Latham, D. W. et al. 2008, ApJ, submitted (arXiv:0812.1161)

Loeb, A., 2005, ApJL, 623 L45

Loeillet, B. et al., 2008, A&A, 481, 529

Lupton, R. 2007, ASP Conf. Ser., 371, 160

Lyne, A. G., Bailes, M. 1992, Nature, 355, L213

Mandel, K., Agol, E., 2002, ApJ, 580, L171

Mandushev, G. et al., 2005, ApJ, 621, 1061

Marcy, G. W. & Butler, R. P., 1992, PASP, 104, 270

McCullough, P. R. et al., 2005, PASP, 117, 783

McCullough, P. R. et al., 2006, ApJ, 648, 1228

Mayor, M., Queloz, D., 1995, Nature, 378, 355

Mink, D. J., 2002, ASP Conf. Ser, 281, 169

Mochejska, B. J., Stanek, K. Z., Sasselov, D. D., Szentgyorgyi, A. H. 2002, AJ, 123, 3460

Mochejska, B. J., Stanek, K. Z., Sasselov, D. D., Szentgyorgyi, A. H., Adams, E., Cooper, R. L., Foster, J. B., Hartman, J. D., Hickox, R. C., Lai, K., Westover, M., Winn, J. N. 2006, AJ, 131, 1090

Moutou, C. et al. 2009, A&A, submitted (arXiv:0902.4457)

Murray, C. D. and Dermott, S. F., 1999, Solar System Dynamics, Cambridge Univ. Press, Cambridge

Naef, D. et al. 2001, A&A, 375, 27

Noyes, R. W. et al., 2008, ApJ, 673, L79

O'Donovan, F. T. et al., 2007, ApJ, 662, 658

Ochsenbein, F., Bauer, P., & Marcout, J. 2000, A&A, 143, 23

Pál, A., Bakos, G. Á., 2006, PASP, 118, 1474

Pál, A. et al., 2008, ApJ, 680, 1450

Pál, A., 2008, MNRAS, 390, 281

Pál, A., Bakos, G. Á., Noyes, R. W., Torres, G., 2008, IAU Symposium, Volume 253, 428

Pál, A. et al. 2008, ApJ, submitted (arXiv:0810.0260)
   An analytical solution for Kepler's problem

Pál, A., 2008, MNRAS, accepted (arXiv:0904.0324)

Pepper, J., Gould, A. & DePoy, D. L. 2004, AIP Conf. Proc. 713: The Search for Other
   Worlds, 713, 185

Pepper, J. et al., 2007, PASP, 119, 923

Perryman, M. A. C., et al., 1997, A&A, 323, 49

Phillips, A. C., Davis, L. E. 1995, in Astronomical Data Analysis Software and Systems IV,
   ASP Conference Series Vol. 77, 297 (eds. R. A. Shaw, H. E. Payne and J. J. E. Hayes)

Pojmanski, G. 1997, Acta Astronomica, 47, 467

Pollacco, D. et al., 2006, Ap&SS, 304, 253

Press, W. H., Teukolsky, S. A., Vetterling, W.T., Flannery, B.P., 1992, Numerical Recipes
   in C: the art of scientific computing, Second Edition, Cambridge University Press

Queloz, D. et al., 2001, A&A, 379, 279

Rauer, H., Eislffel, J., Erikson, A., Guenther, E., Hatzes, A. P., Michaelis, H. & Voss, H.
   2004, PASP, 116, 38

Sahu, K. C., 2006, Nature, 443, 534

Santos, N. C. et al. 2002, A&A, 392, 215

Shewchuk, R. J. 1996, in Applied Computational Geometry: Towards Geometric Engineer-
   ing, ed. M. C. Lin & D. Manocha (Berlin: Springer), 1148, 203

Seager, S., Malln-Ornelas, G. 2003, ApJ, 585, 1038

Shporer, A., Mazeh, T., Moran, A., Bakos, G. Á., Kovács, G. & Mashal, E., 2006, Tenth
   Anniversary of 51 Peg-b: Status of and prospects for hot Jupiter studies, eds. L. Arnold,
   F. Bouchy and C. Moutou (Frontier Group, Paris), 196

Shporer, A. et al., 2009, ApJ, 690, 1393

Shupe, D. L., Moshir, Mehdrdad, Li, J., Makovoz, D., Narron, R., Hook, R. N., 2005, ASP Conf. Ser., 347, 491[2]

Simon, A., Szatmáry, K. & Szabó, Gy. M., 2007, A&A, 470, 727

Skrutskie, M. F., Cutri, R. M., Stiening, R., Weinberg, M. D., Schneider, S., Carpenter, J. M., Beichman, C., Capps, R., Chester, T., Elias, J., Huchra, J., Liebert, J., Lonsdale, C., Monet, D. G., Price, S., Seitzer, P., Jarrett, T., Kirkpatrick, J. D., Gizis, J., Howard, E., Evans, T., Fowler, J., Fullmer, L., Hurt, R., Light, R., Kopan, E. L., Marsh, K. A.,, McCallon, H. L., Tam, R., Van Dyk, S., and Wheelock, S., 2006, AJ, 131, 1163

Snellen, I. A. G. et al. 2008, A&A, accepted (arXiv:0812.0599)

Southworth, J., Wheatley, P. J., & Sams, G. 2007, MNRAS, 379, 11

Sozzetti, A. et al., 2007, ApJ, 664, 1190

Steffen, J. H. & Agol, E., 2007, ASP Conf. Ser, 366, 158

Stetson, P. B., 1987, PASP, 99, 191

Stetson, P. B. 1989, in Advanced School of Astrophysics, Image and Data Processing/Interstellar Dust, ed. B. Barbury, E. Janot-Pacheco, A. M. Magalhães and S. M. Viegas (São Paulo, Instituto Astrônomico e Geofísico)

Stiefel, E. L., Scheifele, G., 1971, Linear and regular celestial mechanics, perturbed two-body motion, numerical methods, canonical theory, Berlin, New York, Springer-Verlag

Street, R. A. et al., 2003, ASP Conf. Ser. 294, 405

Szulágyi, J., Kovács, G. & Welch, D. L. 2009, A&A, accepted (arXiv:0903.1165)

Takeda, G., Kita, R. & Rasio, F. A. 2008, ApJ, 683, 1063

Tamuz, O., Mazeh, T. & Zucker, S. 2005, MNRAS, 356, 1466

Thiebaut, C., Boër, M. 2001, ASP Conf. Ser., 238, 388

Tomaney, A. B., Crotts, A. P. S. 1996, AJ, 112, 2872

Torres, G., Boden, A. F., Latham, D. W., Pan, M. & Stefanik, R. P., 2002, AJ, 124, 1716

Torres, G., Konacki, M., Sasselov, D. D., & Jha, S., 2005, ApJ, 619, 558

Torres, G. et al., 2007, ApJ, 666, L121

[2]http://spider.ipac.caltech.edu/staff/shupe/distortion_v1.0.htm

Torres, G., Winn, J. N., Holman, M. J., 2008, ApJ, 677, 1324

Udalski, A., Szymanski, M., Kaluzny, J., Kubiak, M., Krzeminski, W., Mateo, M., Preston, G. W., Paczynski, B. 1993, AcA, 43, 289

Udalski, A., Paczynski, B., Zebrun, K., Szymanski, M., Kubiak, M., Soszynski, I., Szewczyk, O., Wyrzykowski, L., Pietrzynski, G. 2002, AcA, 52, 1

Valdes, F. G., Campusano, L. E., Velásquez, J. D., Stetson, P. B. 1995, PASP, 107, 1119

Valenti, J. A. & Piskunov, N., 1996, A&AS, 118, 595

Valenti, J. A. & Fischer, D. A., 2005, ApJS, 159, 141

van Leeuwen, F., 2007, Hipparcos, the New Reduction of the Raw Data, Astrophysics and Space Science Library, Vol. 250

van Leeuwen, F., 2007, A&A, 474, 653

Vogt, S. S., 1987, PASP, 99, 1214

Vogt, S. S. et al., 1994, Proc. SPIE, 2198, 362

Weldrake, David T. F., Bayliss, Daniel D. R., Sackett, Penny D., Tingley, Brandon W., Gillon, Michal, Setiawan, Johny 2008, ApJ, 675, 37

Winn, J. N. et al. 2007, ApJ, 665, 167

Winn, J. N. et al. 2007, ApJ, 134, 1707

Wolszczan, A., Frail, D. A., 1992, Nature, 355, 145

Wright, J. T. 2005, PASP, 117, 657

Yi, S. K. et al., 2001, ApJS, 136, 417

Yuan, F. & Akerlof, C. W. 2008, ApJ, 677, 808

Young, A. T., 1967, AJ, 72, 747