

Focal plane geometry characterization of the *Kepler Mission*

Peter Tenenbaum^{*a}, Jon M. Jenkins^a

^aSETI Institute/NASA Ames Research Center, M/S 244-30, P.O. Box 1, Moffett Field, CA
USA 94035

ABSTRACT

The *Kepler Mission* focal plane contains 42 charge-coupled device (CCD) photodetectors. Each CCD is composed of 2.2 million square pixels, 27 micrometers on a side, arranged in a grid of 2,200 columns by 1,044 rows. The science goals of the *Kepler Mission* require that the position of each CCD be determined with an accuracy of 0.1 pixels, corresponding to 2.7 micrometers or 0.4 seconds of arc, a level which is not achievable through pre-flight metrology. We describe a technique for determining the CCD positioning using images of the *Kepler* field of view (FOV) obtained in flight. The technique uses the fitted centroid row and column positions of 400 pre-selected stars on each CCD to obtain empirical polynomials which relate sky coordinates (right ascension and declination) to chip coordinates (row and column). The polynomials are in turn evaluated to produce constraints for a nonlinear model fit which directly determines the model parameters describing the location and orientation of each CCD. The focal plane geometry characterization algorithm is itself embedded in an iterative process which determines the focal plane geometry and the Pixel Response Function for each CCD in a self-consistent manner. In addition to the fully automated calculation, a person-in-the-loop implementation was developed to allow an initial determination of the geometry in the event of large misalignments, achieving a much looser capture tolerance for more modest accuracy and reduced automation.

Keywords: Kepler Mission, ccd, alignment, focal plane geometry

1. INTRODUCTION

The *Kepler Mission*^{1,2} is a space-based photometer which uses transit photometry to search for exoplanets in general, and Earth-size planets located in their stars' habitable zones in particular. The *Kepler* photometer continually monitors a region of the sky of approximately 115 square degrees via a 96 megapixel charge-coupled device (CCD) array. The focal plane array consists of 42 CCDs arranged in a grid with four-way symmetry, such that the field of view is almost undisturbed by the quarterly rotations of the spacecraft necessary to maintain the correct orientation of the solar panels and thermal radiator. Each CCD is arranged in a rectangular pixel grid of 2,200 columns and 1,044 rows and contains two output channels, with 1,100 columns read out by each channel. Each pixel is square, 27 micrometers on a side, and covers an area of sky approximately 3.98 arc-seconds on a side. In order to detect transits, which have typical durations comparable to one day, the photometer uses an exposure interval of 30 minutes.

Due to data volume and bandwidth constraints, it is not possible to store, downlink, or analyze the full 96-megapixel focal plane dataset at 30-minute cadence intervals. Instead, a subset of pixels is selected which provides coverage of the stellar targets regarded as most promising from the point of view of mission science goals. A total of approximately 5.4 million pixels are included, which cover over 160,000 target stars;³ on average, each target is allotted about 33 pixels, with brighter targets given more pixels and dimmer targets given fewer pixels in order to maximize the signal-to-noise performance of each target.

One prerequisite for such a miserly allotment of pixels per target is accurate knowledge of the pixel coordinates for each target star in all four rotated orientations ("seasonal rolls") of the spacecraft. A prerequisite for such accurate knowledge, in turn, is accurate knowledge of the positions of the CCDs on the focal plane: the mission

^{*}peter.tenenbaum@nasa.gov

requirement is a positioning accuracy of 0.1 pixels.⁴ Such an accuracy cannot be met *ab initio*: the achievable accuracy of the CCD array positioning from ground measurements is approximately 3 pixels.⁵ A method is therefore required for performing on-orbit determination of the CCD positions on the focal plane array with the necessary 0.1-pixel accuracy.

The method of on-orbit CCD alignment developed for the *Kepler Mission* makes use of images of the *Kepler* field of view, as captured by the photometer, to determine the locations of the CCDs on the focal plane. In particular, the known right ascension (RA) and declination (Dec) of the stars in the field of view are used to constrain a nonlinear fit of the CCD positions. This technique and the software used to implement it are known collectively as Focal Plane Geometry (FPG) characterization.

2. FIT CONSTRAINTS AND PARAMETERS

The focal plane geometry characterization software makes use of two mappings between sky coordinates (in right ascension and declination) and pixel coordinates (in output channel, row, and column). The first of these is based on a predictive model; the second is an empirical mapping, in which a polynomial fit is performed on image data obtained in flight. Both mappings are produced by MATLAB⁶ software which is part of the *Kepler* data analysis codebase. The ability to leverage the existing *Kepler* analysis codebase dramatically reduced the amount of developer time required to implement the FPG characterization software.

2.1 Focal Plane Characterization

The SOC codebase includes a predictive, model-based tool for converting sky coordinates to pixel coordinates and vice versa.⁷ The tool, known as *raDec2Pix*, is implemented as a MATLAB class. The *raDec2Pix* class requires the following information to perform the coordinate conversion:

- The spacecraft *absolute* pointing at the time of interest
- The spacecraft velocity at the time of interest
- The spacecraft seasonal rotation at the time of interest
- The plate scale of the photometer optics
- The center coordinates of each CCD in the focal plane.

In the standard usage of *raDec2Pix*, all of the necessary items in the above list are provided to the software via models which are stored in a database and extracted for use at the moment the *raDec2Pix* object is instantiated: a pointing model, roll-time model, and geometry model provide the first, third, fourth, and fifth items on the list; the spacecraft velocity is obtained via use of the *SPICE* toolkit,⁸ which is directly incorporated into *raDec2Pix* via a MATLAB *mexfile*. An alternate usage permits the user to supply the spacecraft absolute pointing, if the pointing model's accuracy is suspect. The standard usage includes the effect of velocity aberration, though this may be suppressed by the user.

The geometry model includes the positioning and plate scale of each CCD in the focal plane. The position information is stored in the form of *3-2-1 transformation* angles;⁹ thus, the model does not encode the CCD position in pixels or micrometers, but rather in the angle offset of the CCD's field of view from the boresight pointing.

2.2 Motion Polynomials

Each CCD in the *Kepler* focal plane contains an average of about 4,000 target stars (with more target stars on the CCDs closer to the plane of the Milky Way, and fewer target stars on CCDs further away). Of these, about 400 stars per CCD are chosen for use in monitoring the photometer performance over time. The 400 stars chosen this purpose are bright, unsaturated stars without known nearby stars ("crowding"), since stars with these characteristics have been shown to be optimal for the purpose of photometer performance assessment.¹²

Performing a fit with 4,000 or even 400 point-constraints per CCD would be computationally prohibitive. Instead, a polynomial representation of the star positions is derived: the star centroid positions in pixel coordinates are determined via a fit of the star images to the relevant pixel response functions,¹⁰ and two-dimensional polynomials are fitted which relate the centroid positions in pixel coordinates to the star positions in sky coordinates: $R = P_R(RA, Dec)$, $C = P_C(RA, Dec)$. The polynomial fits are robust, using standard techniques to de-emphasize the contribution of outliers in the calculation of the polynomials. The correct polynomial order is determined independently for each polynomial (row and column, for each output channel) via Akaike's Information Criterion.¹³ The resulting polynomials (somewhat confusingly named *motion polynomials*¹¹) are used by the FPG characterization to provide the constraints to the fit of CCD positions.

Constraining the FPG fit via the motion polynomials has several advantages over constraining the fit with the fitted star centroid positions used to construct the motion polynomials. First, the positions on each CCD that are used to provide fit constraints can be chosen for convenience regardless of the presence or absence of actual, physical stars at the desired location. Second, the impact of outliers has already been reduced due to the use of robust polynomial fitting in the generation of the motion polynomials. Third, and most important, the polynomial coefficients are computed using the full set of monitoring stars for each output channel, minus outliers; therefore the covariance matrix for the polynomial coefficients is small, as it benefits from the full statistical power of the large number of monitoring stars on each output channel. This, in turn, means that when the motion polynomials are evaluated to determine which row and column correspond to a particular RA and Dec, the uncertainties in the evaluated row and column values are much smaller than the row and column uncertainties present on the centroid fits of the individual stars on each CCD. In short, use of the motion polynomials gives FPG the statistical benefits of a large number of stars constraining the position of each CCD, without obliging FPG to suffer the computational disadvantage of performing a fit that includes a large number of constraints.

2.3 Overall Approach

Given the two mappings between sky and pixel coordinates described above, the most sensible fit procedure for the focal plane geometry is to select a set of points in the sky, use both the empirical and the model converters to convert the selected sky points to coordinates on the focal plane, and adjust the geometry parameters in the model converter until the differences between the two mappings are minimized. The sky coordinates for use in this process are determined by selecting a rectangular grid of points on each CCD and converting from pixel to sky coordinates using *raDec2Pix*. This conversion must be performed prior to the fit, and therefore requires the use of the pre-fit geometry model in *raDec2Pix*, which can contain errors of up to three pixels relative to the true positions of the CCDs. This error is not worrisome, since the sole purpose of this conversion is to select sky coordinates that provide good coverage of the focal plane, and for this purpose approximate knowledge of the location of the CCDs is sufficient. In order to avoid the possibility that a selected point falls between CCDs, on a different CCD, or on a different output channel from the intended one, points are selected that are at least 100 pixels from any edge of a CCD or from the boundary between readout channels on a CCD.

The fit process fits three geometry parameters for each CCD, corresponding to the three angles of the 3-2-1 transformation, plus one plate scale parameter for the entire field of view; in order to fully constrain the fit, therefore, at least two points are needed on each CCD, since each point has two coordinates (row and column), and therefore two points will provide a total of four constraints. At the other extreme, the maximum number of constraint points is limited by the practical requirement of acceptable computation time, and also by the theoretical limits implied by the motion polynomial fit: a two-dimensional polynomial of order N has $(N+1)(N+2)/2$ coefficients, therefore evaluating the motion polynomial at more than $(N+1)(N+2)/2$ points does not provide additional independent constraints for the FPG fit. In normal use, the FPG fitter is configured to use eight points per CCD, or four points per output channel, to constrain the fit, with the eight points set in a 4×2 rectangular grid and all points at least 100 pixels from any edge. With four constraint points per output channel, the polynomial order degeneracy described above is not an issue for any motion polynomial of order two or more, and the projection of the curved sky coordinate system onto the CCDs guarantees that motion polynomials must be at least second order.

With three geometry parameters per CCD, 42 CCDs, and one plate scale parameter for the field of view, the total number of geometry parameters in the fit is 127. The total number of constraints is eight points per CCD,

with two constraints (row and column) per point, for a total of 672. The number of degrees of freedom in the fit is therefore 545.

2.3.1 Fitting spacecraft pointing

As described in Section 3, below, the FPG characterization fit can be performed using motion polynomials generated from images obtained at more than one time. In this case, it is both necessary and straightforward to include the spacecraft pointing in the fit. For images taken at M times, one time is selected as the reference time and the pointing at that time is assumed to be the nominal pointing (or, more specifically, the pointing in the pointing model used by *raDec2Pix*); for the other $M - 1$ times, the RA, Dec, and rotation of the spacecraft become additional fit parameters. In this case, the Levenberg-Marquardt fitter varies both the focal plane geometry parameters in the *raDec2Pix* model and the pointing for the $M - 1$ images taken at non-reference times in order to minimize the difference between the model pixel coordinates and the pixel coordinates obtained from the motion polynomials.

In the case of an FPG characterization fit using images from only one time, fitting the CCD positions and the pointing simultaneously introduces a degeneracy: moving the spacecraft pointing in one direction cannot be distinguished from systematically moving the CCD positions by the same amount in the opposite direction. In order to perform a simultaneous fit of the pointing and focal plane geometry with only one set of images, an additional set of constraints is required to break this degeneracy. The intended definition of the spacecraft pointing is that the line of the spacecraft pointing passes through the center of gravity of the CCD array, and the rotation angle of the pointing is such that the arrays have no net rotation with respect to the spacecraft coordinate system. With this definition, it was straightforward to define a constraint on the sum of the “3” angles, a constraint on the sum of the “2” angles, and a constraint on the sum of the “1” angles such that the fitted spacecraft pointing was consistent with the intended definition of same. The additional constraints were weighted such that a change to the spacecraft pointing parameters would change the value of χ^2 exactly as much as making the same change systematically to all of the CCD positions.

2.3.2 Robust fitting

The FPG characterization fitter uses the MATLAB function *nlinfit* to perform its Levenberg-Marquardt nonlinear optimization.^{14,15} The *nlinfit* function supports robust fitting: an initial fit is performed, and the residuals for each data point are determined; a weighting is applied which increases the prominence of points with small residuals and decreases the prominence of points with large residuals; the fit is repeated, making use of the weights determined after the previous fit iteration; weighting and fitting are iterated until convergence is achieved.

The standard convergence criterion for *nlinfit* is the change in parameter values: specifically, when all parameters change by less than ϵ of their values from one iteration to another, convergence is declared. For double-precision real parameters, *nlinfit* defines $\epsilon \equiv 1.49 \times 10^{-8}$, which is an extremely tight convergence criterion. This is especially true for parameters that converge to values close to zero, since such parameters may vary by amounts that are comparable to their absolute values but small compared to their fit uncertainties – in other words, for such parameters *nlinfit* may be unable to converge despite the fact that it is no longer changing their values significantly. To address this, a modified version of *nlinfit* was developed which defines convergence as occurring when all parameters change by less than δ of their fit uncertainties from one iteration to another. The convergence parameter δ was added to the set of user-configurable parameters in this version of *nlinfit*, and set to 0.5 for FPG characterization fits. This improved convergence speed considerably, with negligible impact in the quality of convergence.

2.3.3 Other options

Although the nominal usage of the FPG characterization fitter is on datasets that cover the full focal plane, several contingencies were envisioned in which this would be either impossible or undesirable. Consequently, the fitter allows the user to specify output channels that are not to be included in the fit. In the event that both channels on a given CCD are omitted, the CCD’s geometry parameters are excluded from the fit and the constraint on the CCD center of mass used to break the pointing-geometry degeneracy is appropriately modified. The FPG characterization fitter also allows the user to specify a data completeness level below which fitting will not be attempted.

In addition to options that used the *raDec2Pix* pointing model and fitted the pointing, the FPG characterization fitter allows the user to specify an absolute pointing to use.

The user can choose to either fit the plate scale or else to accept the current plate scale and fit only the CCD positioning parameters.

3. IMPLEMENTATION AND USE CASES

The Focal Plane Geometry characterization algorithm was implemented as a standard *Kepler* science processing module, with all numerical and algorithmic content written in MATLAB and all supporting infrastructure such as communication with the database and filestore written in Java.²⁰ The MATLAB implementation makes use of MATLAB's support for object-oriented programming, with one class (*fpgDataClass*) used to manage overall data and execution flow, a second class (*fpgFitClass*) used to manage the Levenberg-Marquardt fit, and a third class (*fpgResultsClass*) used to package results and generate graphics. The *fpgResultsClass* is a child class of the *fpgFitClass*, while the *fpgDataClass* contains one or more embedded *fpgFitClass* objects but does not inherit their methods or members.

For *Kepler Mission* commissioning, a total of three use cases for the FPG fitter were identified, each of which presented its own requirements and challenges in terms of configuration and data handling. The use cases are described below.

3.1 Single Dataset Fitting

Determination of the focal plane geometry was expected to be an extremely early operation after acquisition of first light, since knowledge of the geometry with an accuracy better than the pre-flight level of three pixels was required for many commissioning activities. It was therefore necessary to be able to perform FPG characterization using one dataset acquired simultaneously on all 42 CCDs.

Given the relatively poor knowledge of the focal plane geometry at first light, and the risk that the three-pixel expected accuracy might not be achieved for one or more CCDs, a decision was taken to perform single-dataset FPG characterization using a full-field image (FFI), which includes all 96 megapixels, rather than using one of the standard datasets which contains only 5.4 megapixels; due to the criticality of the FPG fit for other commissioning activities, the first-light FFI was selected for use in single-dataset FPG fitting. The pre-processing steps required prior to FPG characterization make use of *Kepler* processing algorithms that are only supported for standard datasets: specifically, these pre-processing steps include image calibration¹⁶ and photometric analysis,¹¹ the latter of which produces the motion polynomials required as an FPG input. In order to manage this apparent contradiction, a Java module was written which takes as input an FFI and a target pixel definition file and produces as output a standard *Kepler* dataset containing the requested pixels from the FFI, packaged in the correct format for ingestion by the *Kepler* processing system. If FPG analysis revealed that one or more CCDs was further from its nominal position than the expected three pixels, the process could be repeated with an updated definition of target pixels for the FFI extraction step. The risk of such an outcome was further reduced by selecting a cluster of pixels for each target that was larger than would be used in nominal science processing, at the expense of reducing the number of targets available by about a factor of 2.

For the target stars used in this process, star centroids were determined by waveform fitting to a point spread function that had been modified to include the effects of sub-pixel response variation and blurring due to spacecraft pointing jitter, also known as a *pixel response function*.¹⁰ The pixel response functions (PRFs) used were pre-flight estimates based on ray-tracing through the *Kepler* as-designed optics, and not PRFs determined empirically from flight data. The imperfect knowledge of the actual PRF placed an additional limitation on the expected accuracy of the FPG fit using the first-light FFI, but the process was expected to be sufficiently accurate to permit commissioning activities to continue.

3.2 Multiple Dataset Fitting

At a later point in commissioning, the pixel response function waveforms are determined empirically by acquiring 122 datasets, each of which is acquired with a slightly different spacecraft pointing, and performing a least-squares fit in which the parameters are a polynomial representation of the PRF. The PRFs are determined separately for each of the 84 output channels, and each output channel is further subdivided into five regions to allow variation in the PRF across one output channel to be taken into account. A full description of the PRF characterization and use can be found elsewhere.¹⁰

In order to perform an accurate determination of the PRF, the mapping from sky to pixel coordinates must be accurate as well, which implies that the focal plane geometry model must be as accurate as possible. As described in the previous section, however, a key limitation to the accuracy of the FPG fit is the limited accuracy of the PRF waveforms are used to perform the centroiding fit on the target stars. Therefore, the procedure for fitting the PRF waveforms includes invocation of the FPG fit procedure. More specifically, the two fits – PRF and FPG – are performed sequentially, and iteratively, until both fits have converged upon a self-consistent set of PRF waveforms and CCD positions. The main metric used to evaluate convergence is the iteration-to-iteration change in fitted CCD positions.

It is important to note that, when performing a fit of the PRF waveforms and the CCD positions as described above, there is a degeneracy due to the indeterminacy of the center of the waveform: a fit of the waveform and the CCD position cannot be distinguished from a fit to the same data in which the waveform and the CCD position are both shifted by the same amount in the same direction. This degeneracy was corrected by constraining the position of the PRF waveform in the fit.

The FPG fitter was also responsible for determining the actual attitude of the spacecraft for each dataset other than the first, which was acquired at the nominal Kepler science attitude. In early tests of the multiple-dataset fit, the performance of the fitter was found to be completely unacceptable, and an improved procedure was developed:

- The focal plane geometry was fitted using only the first dataset of the 122 datasets collected for PRF determination, with an initial estimate given by the FPG fit from single-dataset fitting (see Section 3.1)
- The spacecraft attitude for each of the 121 remaining datasets was fitted, using the CCD positions fitted using the first dataset of the 122 datasets collected for PRF determination
- The CCD positions and the spacecraft attitudes were fitted simultaneously, using the results of preceding fits as initial estimates of the fit parameter values.

In the final fit, which simultaneously fits the CCD positions and the spacecraft attitudes, the total number of fit parameters is 490 (42 x 3 CCD positions, 1 plate scale, 121 x 3 attitude parameters), and the total number of constraints is 81,984 (42 CCDs x 8 positions x 2 coordinates x 122 datasets). The 122-dataset fit would converge in approximately two hours when run on a modern (2009 model) computer.

3.3 Person-in-the-Loop Fitting

The use cases described above are all fully automated, operating in the context of the *Kepler* analysis pipeline,¹⁷ and rely on the motion polynomials rather than actual image data. While such an approach is appropriate and valid for an operational environment in which major surprises have been ruled out, there are benefits to including human supervision and judgement during the very earliest stages of commissioning, when such surprises are much more likely. For these reasons, a person-in-the-loop version of the FPG fitter was developed, based on the existing FPG codebase and the MATLAB GUI toolkit.

Figure 1 shows a screen capture of the person-in-the-loop FPG fitter’s main GUI. The GUI displays the pixels from one CCD, which are extracted from an FFI, and also displays the expected positions of stars on the CCD, using the *Kepler Input Catalog* (KIC)¹⁸ and *raDec2Pix* (green circles in Figure 1); the user may display stars based on a selected range of stellar magnitudes. The user then employs mouse clicks to select target stars and to connect the model positions of target stars (blue circles in Figure 2) to the actual positions of those stars in the

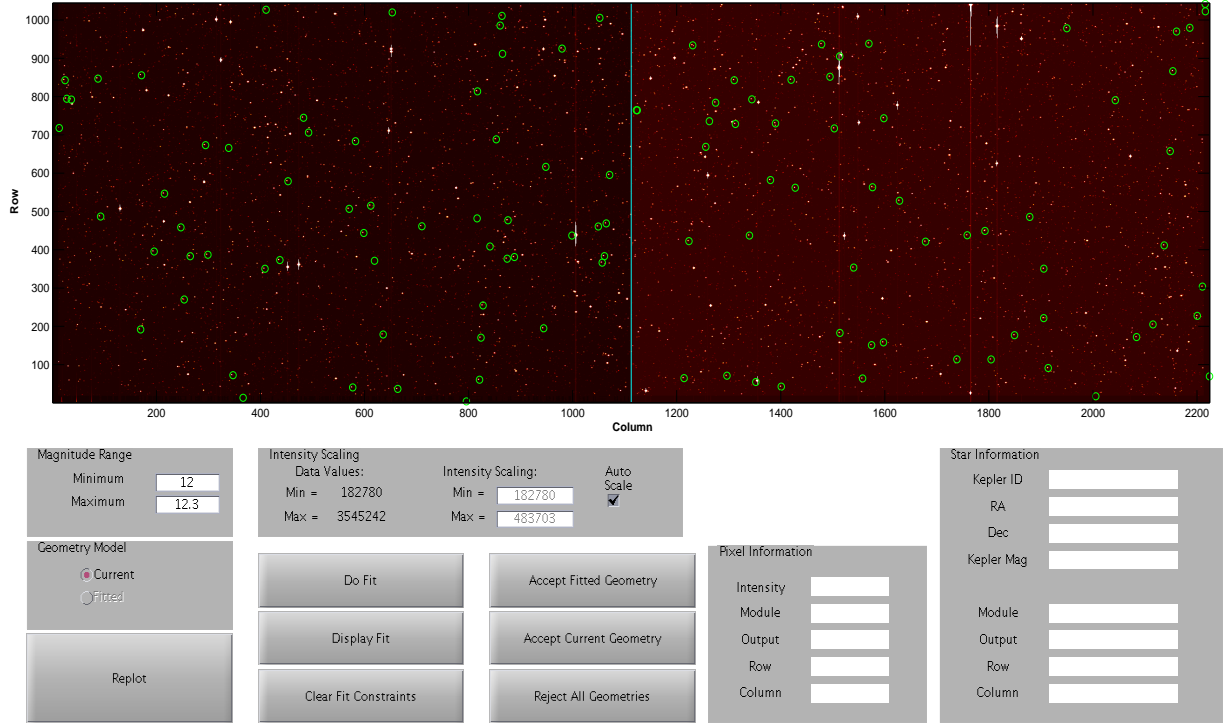


Figure 1: Person-in-the-loop fitter main GUI. The CCD image is displayed, along with the model positions of stars in a selected magnitude range (green circles). In this example, model positions for stars with magnitudes from 12.0 to 12.3 are shown.

FFI. Once sufficient stars are selected as constraints, a fit is performed of the CCD's position in the focal plane. A single CCD can be fitted and re-fitted multiple times, as the user can elect to use the result of a previous fit as the starting estimate of the CCD position for an additional fit.

The person-in-the-loop fitter has a number of limitations which make it unacceptable for production of the final focal plane geometry model. First, the fitter does not allow the determination of the plate scale, but rather requires that the fit accept the plate scale in the pre-flight focal plane geometry model. Second, since the user can only specify the pixel that contains each star's true location, sub-pixel accuracy is out of the question. Third, each CCD is fit independently and with human intervention, leading to a relatively long execution time for completing the full focal plane: in tests, users were able to fit all 42 CCDs in about two hours once they had enough practice to be familiar with the GUI. The main advantage of the person-in-the-loop fitter is that it allows a human to see all the pixels in the focal plane, and to see the model locations of stars overlaid on the image; this allows quick diagnosis of problems that would potentially cause automated fitting to fail. The other advantage is that, unlike the automated fitting methods, the person-in-the-loop fitter will run successfully even for extremely large offsets of the CCDs with respect to their expected positions. In the event of an extremely large offset, the user selects the brightest stars on the CCD for use in constraining the fit; since the brightest stars on a CCD form a unique pattern which is easy for the human eye to detect, this approach allows even a severely misaligned CCD to be correctly fitted to within about one pixel of its true position. Once the person-in-the-loop fitter has been used to determine all CCD positions to this level, the resulting focal plane geometry model is then ingested into Kepler DB¹⁹ and used as the starting point for the single-dataset automated fit described in Section 3.1.

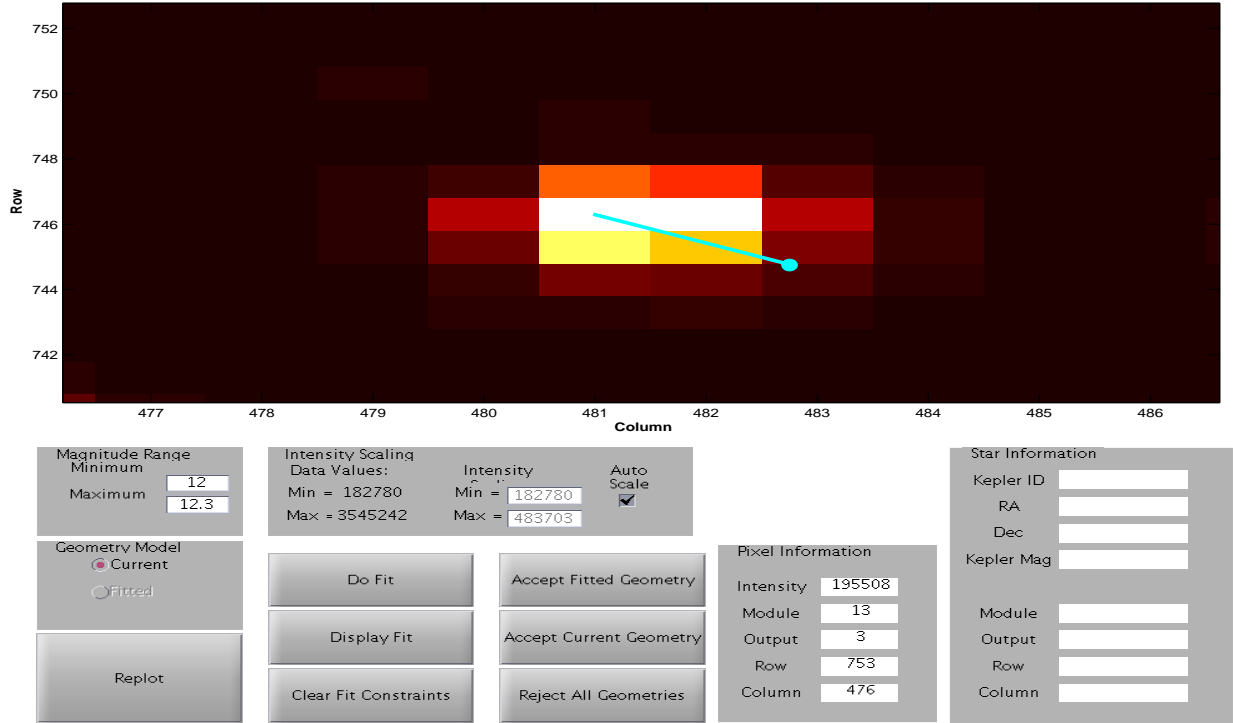


Figure 2: Selecting a constraint for the person-in-the-loop fitter main GUI. The user zooms in on a small region of the image, then employs mouse clicks to select a star based on its model position (cyan circle), and to connect it to the pixel which contains the center of that star in the image (cyan line).

4. OPERATIONAL EXPERIENCE

The *Kepler Mission* spacecraft launched on March 6, 2009, with first light exposure occurring on April 8, 2009 and start of mission science on May 12, 2009.²¹ During the period from April 8 to May 12, all of the use cases described in Section 3 were exercised.

4.1 Person-in-the-Loop Fitter

The GUI-based FPG fitter was used on two occasions. The fitter was first used with an FFI which was acquired while the spacecraft was still in “coarse point,” in order to get an early indication of any problems with the focal plane alignment. Although the spacecraft pointing differed from the nominal by over 60 pixels, it was straightforward to use the GUI to eliminate the bulk offset due to the pointing and to determine that there were no excessive misalignments within the focal plane.

Subsequent to the “coarse point” FFI, the GUI-based FPG fitter was used again with the first FFI acquired in “fine point.” It was quickly observed that there was a systematic offset in the CCD positions relative to expected, which indicated that the spacecraft attitude was off-nominal. The magnitude of the offset was approximately 2.5 pixels. The cause of the discrepancy was hypothesized to be due to a difference between the center of gravity of the 42 science CCDs and the 4 fine guidance sensors (FGSs), which share the focal plane with the science CCDs. In the interest of operational simplicity, a decision was taken to define the pointing obtained by the FGS CCDs as the nominal pointing, and to absorb the systematic difference between FGS and science CCDs into the focal plane geometry model of the science CCDs.

The geometry model produced via the GUI and human-in-the-loop fitting was used as a seed for the single-dataset automated fitting, and this step in the procedure went smoothly.

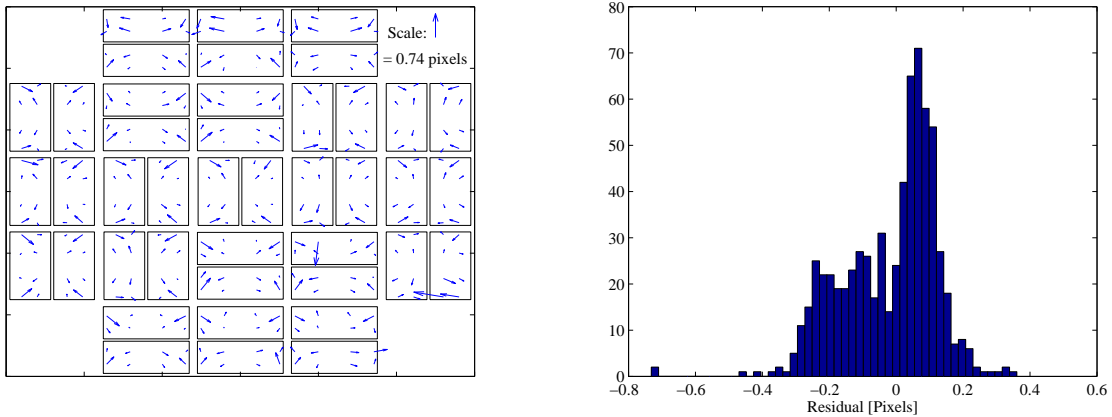


Figure 3: (Left) Quiver plot of the fit residuals from automated fitting. Note the systematic shape to the distribution of residuals on each module. (Right) Distribution of the fit residuals from automated fitting. Note the bi-modal shape of the distribution.

4.2 Multiple Dataset Fitting

Subsequent to the automated single-dataset fit, FPG was next used in the FPG-PRF fitting loop, which analyzed 122 datasets acquired over three days. Figure 3 shows a quiver plot of the fit residuals from this procedure for a single dataset, and a histogram of fit residuals for all 122 datasets. Together, these plots indicate that *Kepler's* optical system contains properties that were not included in the focal plane geometry model, and which result in fit residuals over the 0.1 pixel requirement.

The pattern of residuals in Figure 3 suggested that the remaining optical effect is at the level of CCD pairs (known as a readout *module*). Each readout module has a lens that corrects for the shape discrepancy between the CCDs, which are flat, and the focal plane of *Kepler's* Schmidt optics, which is curved; the optics of these “field-flattener” lenses was an obvious suspect. Further analysis indicated the need for two changes to the fitting algorithm. First, a unique plate scale was required for each CCD, rather than accepting a single fitted plate scale for the entire focal plane. Second, a pincushion correction was required for each CCD, to take into account the observation that the plate scale is smaller at the center of a module than it is at the periphery. If we consider a coordinate system which is centered at the center of a module, in which X and Y are the offsets from the center in arc-seconds, the conversion to offsets in row and column, R and C , is given by:

$$R = X/P_1(1 + P_2r^2), \quad (1)$$

$$C = Y/P_1(1 + P_2r^2), \quad (2)$$

where P_1 is the plate scale in arc-seconds per pixel, P_2 is the pincushion correction in inverse square arc-seconds, and r is the radius of the offset in arc-seconds, $r^2 = X^2 + Y^2$. The magnitude of P_2 was found to be around 2.6×10^{-11} inverse square arc-seconds for typical modules; omission of this factor, based on Equation 1, would result in a conversion error of almost one full pixel in each of row and column for points in the corners of a module.

One significant disadvantage to the additional fit parameters is that the execution time for the 122 dataset FPG fit expanded from 2 hours to approximately 12 hours. In this configuration, the total number of fit parameters is expanded from 490 to 574.

Figure 4 shows the fit residuals in for the FPG constraint points after a fit including the pincushion correction and separate plate scales for each CCD. Comparison to Figure 3 shows that the changes made to the plate scale and pincushion fitting have eliminated the largest of the observed systematic distortions. The maximum residual remaining is slightly over 0.1 pixels. The shape of the distribution suggests that some other, less prominent

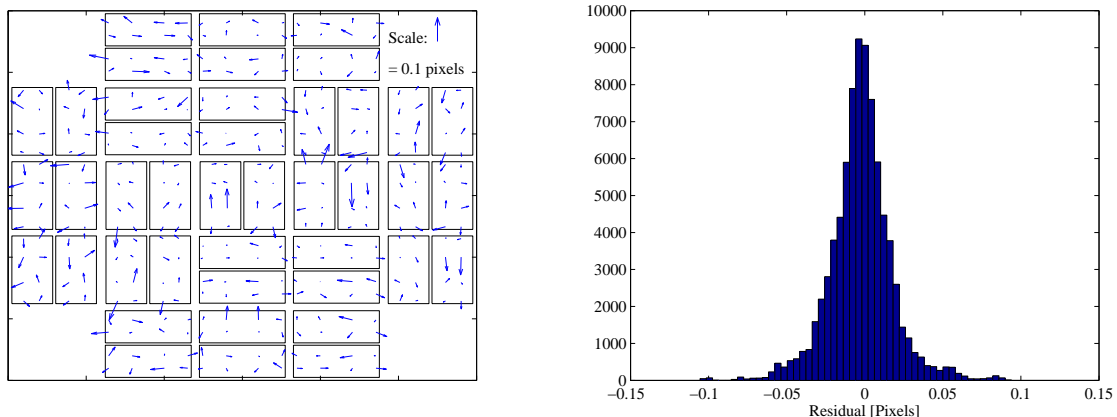


Figure 4: Fit residuals after including the pincushion correction and fitting a separate plate scale for each CCD. (Left) Quiver plot of the fit residuals. (Right) Distribution of the fit residuals.

systematic distortions are still present in the optics, but at an acceptably low level. Figure 5 shows the plate scale and pincushion corrections for each of the 42 CCDs. Note that there is no apparent correlation between the fitted values for the two CCDs in a module (CCDs 1 and 2, 3 and 4, etc.). Such a correlation would be expected if the variation in plate scale and pincushion correction were solely due to the optics of the “field-flattener” lenses. It is hypothesized that the placement of the individual CCDs on the focal plane may be responsible for the differences in parameters between the two CCDs which nominally share the same optics.

Figure 6 shows the fitted offsets for the 42 CCDs relative to the pre-flight focal plane geometry model, and also shows the estimated uncertainties in the fitted positions. The typical CCD positioning accuracy was closer to two pixels than three pixels, and the statistical uncertainties on fitted CCD positions are small compared to the 0.1-pixel requirement. Note that Figure 6 does not include the 2.5-pixel offset described in Section 4.1.

5. CONCLUSIONS

The Focal Plane Geometry characterization software is a pipeline module which uses flight images from the *Kepler* focal plane to determine the as-built positions of the CCDs on the focal plane. Three related implementations of the FPG characterization software were developed and used during *Kepler* commissioning. Although the commissioning data presented a few unexpected characteristics, the FPG software ultimately allowed the CCD positions to be determined with the required 0.1-pixel accuracy. The achieved accuracy has permitted the *Kepler Mission* to allocate relatively few pixels to each science target in the field of view, which in turn has permitted the selection of over 150,000 science targets for the mission.

ACKNOWLEDGMENTS

The excellent performance reported here was the result of a team effort involving the *Kepler* Science Team, Science Office, and Science Operations Center. In particular, the work of Jay Gunter, Jennifer Hall, Khadeejah Ibrahim, Sean McCauliff, Brett Stroozas, Kamal Uddin, and Hayley Wu was indispensable, as was the leadership and participation of Bill Borucki, Mike Haas, and Dave Koch.

Funding for the *Kepler Mission* is provided by NASA’s Science Mission Directorate.

REFERENCES

- [1] Borucki, W., *et. al.*, “Kepler planet-detection mission: introduction and first results,” *Science* 327, 977-980 (2010).

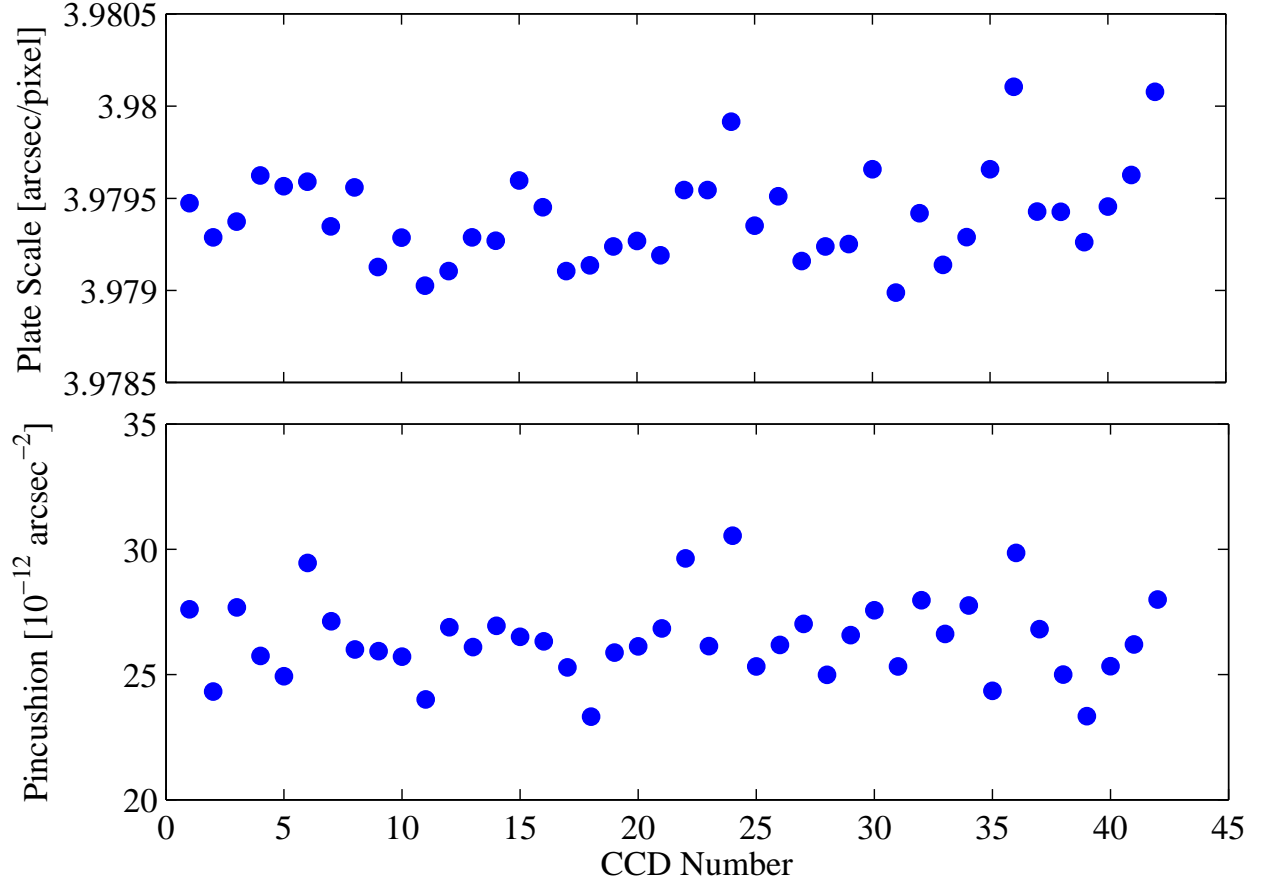


Figure 5: Plate scales (top) and pincushion corrections (bottom) from the automated 122-dataset FPG fit.

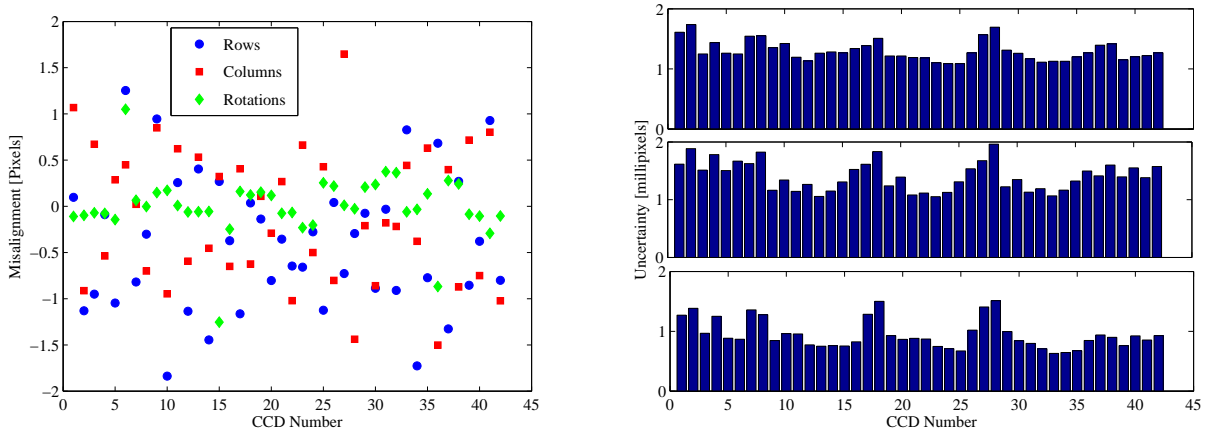


Figure 6: Fitted position offsets of the 42 CCDs with respect to the pre-flight model (left), and bar chart of estimated uncertainties from the fit (right).

- [2] Koch, D. G. *et al.*, “*Kepler* mission design, realized photometric performance, and early science,” *ApJ Lett.* 713(2), L79-L86 (2010).
- [3] Batalha, N. M. *et al.*, “Selection, prioritization, and characteristics of *Kepler* target stars,” *ApJ Lett.* 713(2), L109-L114 (2010).
- [4] Tenenbaum, P., “Software Design Document: Commissioning Tool: Focal Plane Geometry,” unpublished (2008).
- [5] Ortiz, R. A., “Focal Plane Mechanical Tolerancing,” unpublished (2003).
- [6] The Mathworks, MATLAB r2007a (2007).
- [7] Allen, C., Klaus, T., Jenkins, J. M., “*Kepler Mission’s* focal plane characterization models implementation,” *Proc. SPIE 7740*, in press (2010).
- [8] <http://naif.jpl.nasa.gov/naif/toolkit.html> (2009).
- [9] Goldstein, H., [Classical Mechanics], Addison-Wesley Publishing Company, Reading, Mass., 143-166, 608-610 (1980).
- [10] Bryson, S. T., *et al.*, “The *Kepler* pixel response function,” *ApJ Lett.* 713(2), L97-L102 (2010).
- [11] Twicken, J. D. *et al.*, “Photometric analysis in the *Kepler* Science Operations Center pipeline”, *Proc. SPIE 7740*, in press (2010).
- [12] Lie, J. *et al.*, “Photometer performance assessment in *Kepler* science data processing,” *Proc. SPIE 7740*, in press (2010).
- [13] Akaike, H., “A new look at thse statistical model identification,” *IEEE Trans. Automat. Control*, 19(6), 716-723 (1974).
- [14] Levenberg, K., “A method for the solution of certain non-linear problems in least squares,” *Q Appl. Math.*, 2(2), 164-168 (1944).
- [15] Marquardt, D., “An algorithm for least-squares estimation of nonlinear parameters,” *J. SIAM Appl. Math.*, 11(2), 431-441 (1963).
- [16] Quintana, Elisa V. *et al.*, “Pixel level calibration in the *Kepler* science operations center pipeline,” *Proc. SPIE 7740*, in press (2010).
- [17] Klaus, T. *et al.*, “The *Kepler* SOC pipeline configuration and execution,” *Proc. SPIE 7740*, in press (2010).
- [18] Latham, D. W. *et al.*, “The *Kepler* Input Catalog,” *Proc. AAS 207*, 1340 (2005).
- [19] McCauliff, S. D. *et al.*, “The *Kepler* DB: a database management system for arrays, sparse arrays, and binary objects,” *Proc. SPIE 7740*, in press (2010).
- [20] Middour, C. *et al.*, “*Kepler* Science Operations Center architecture,” *Proc. SPIE 7740*, in press (2010).
- [21] Haas, M. R. *et al.*, “*Kepler* Science Operations,” *ApJ Lett.* 713, L92-L96 (2010).