

More Flexibility in Representing Geometric Distortion in Astronomical Images

David L. Shupe^a, Russ R. Laher^b, Lisa Storrie-Lombardi^b, Jason Surace^b, Carl Grillmair^b,
David Levitan^c, and Branimir Sesar^c

^aNASA Herschel Science Center, California Institute of Technology, M/S 100-22, Pasadena,
CA 91125, U.S.A.;

^bSpitzer Science Center, California Institute of Technology, M/S 314-6, Pasadena, CA 91125,
U.S.A.;

^cDivision of Physics, Mathematics, and Astronomy, California Institute of Technology,
Pasadena, CA 91125, U.S.A.

ABSTRACT

A number of popular software tools in the public domain are used by astronomers, professional and amateur alike, but some of the tools that have similar purposes cannot be easily interchanged, owing to the lack of a common standard. For the case of image distortion, SCAMP and SExtractor, available from Astromatic.net, perform astrometric calibration and source-object extraction on image data, and image-data geometric distortion is computed in celestial coordinates with polynomial coefficients stored in the FITS header with the PVi_j keywords. Another widely-used astrometric-calibration service, Astrometry.net, solves for distortion in pixel coordinates using the SIP convention that was introduced by the Spitzer Science Center. Up until now, due to the complexity of these distortion representations, it was very difficult to use the output of one of these packages as input to the other. New Python software, along with faster-computing C-language translations, have been developed at the Infrared Processing and Analysis Center (IPAC) to convert FITS-image headers from PV to SIP and vice versa. It is now possible to straightforwardly use Astrometry.net for astrometric calibration and then SExtractor for source-object extraction. The new software also enables astrometric calibration by SCAMP followed by image visualization with tools that support SIP distortion, but not PV . The software has been incorporated into the image-processing pipelines of the Palomar Transient Factory (PTF), which generate FITS images with headers containing both distortion representations. The software permits the conversion of archived images, such as from the Spitzer Heritage Archive and NASA/IPAC Infrared Science Archive, from SIP to PV or vice versa. This new capability renders unnecessary any new representation, such as the proposed TPV distortion convention.

1. INTRODUCTION

The impetus for this work stemmed from the image-processing requirements of the Palomar Transient Factory (PTF).¹ PTF images contain moderate geometric distortion and every exposure needs to be astrometrically calibrated. However, the FITS World Coordinate System (WCS) standard² does not define how to handle deviations from the celestial projections defined in the so-called Paper II (Ref. 3), and an effort to develop a “Paper IV” on a standard for representing distortion ultimately stalled. It has therefore been left to astronomers and

Further author information: (Send correspondence to D.L.S.)

D.L.S.: E-mail: shupe@ipac.caltech.edu, Telephone: 1 (626) 395 2357

R.R.L.: E-mail: laher@ipac.caltech.edu

L.S.-L.: E-mail: lisa@ipac.caltech.edu

J.S.: E-mail: jason@ipac.caltech.edu

C.G.: E-mail: carl@ipac.caltech.edu

D.L.: E-mail: dlevitan@caltech.edu

B.S.: E-mail: bsesar@astro.caltech.edu

developers of astronomical software to define conventions for handling distortion, which have been implemented in some software packages, but not in others.

There are two special challenges for PTF, and any similar observatory concerned with time-domain astronomy. First, it is critical to preserve the individual images without remapping onto a sky grid, to best provide the ability to detect and measure transient events. Second, for PTF, and any ground-based observatory, the distortion solution changes from one exposure to the next owing to atmospheric refraction. These challenges make it necessary not only for the PTF pipelines to efficiently compute the astrometry and the distortion solution, but also for the resulting data products to be compatible with as many tools as possible.

For PTF, the astrometry and the distortion are computed by SCAMP,⁴ as it not only matches image-extracted source objects to a reference catalog to find an image's world coordinate system (WCS), but it also characterizes the distortion. SCAMP's implementation for representing the distortion in the FITS header is based on an early draft* of the WCS standard, which was later dropped. We refer to this implementation as *PV* distortion because it generates FITS keywords of the form *PVi-j* to store distortion-polynomial coefficients in the image header. This *PV* distortion has also been implemented with other popular general software tools (notably SExtractor⁵ for source-object extraction from FITS images, and SWarp⁶ for resampling and co-adding FITS images) by the same author, Emmanuel Bertin.

A distortion convention that is widely supported in many tools is the SIP (Simple Imaging Polynomial) scheme developed by the Spitzer Science Center (SSC). This convention is perhaps the most widely used method for representing geometric image distortion in FITS headers. Many common FITS-image visualization tools now include SIP distortion when computing sky coordinates, and another very successful astrometric-calibration service, Astrometry.net,⁷ solves for the distortion using the SIP convention. Ref. 8 first introduced the SIP convention and lists the astronomical software programs that were upgraded to apply SIP coefficients through cooperative efforts by the SSC and other institutions (our updated list is given below in §6). Unfortunately, the SCAMP/SExtractor/SWarp suite does not support SIP, yet many popular FITS-image visualization tools do not support the *PV* representation. The objective of the present work is to deal with this shortcoming by providing a converter between the *PV* and SIP representations.[†]

In §2, we review the *PV* and SIP distortion representations, which are widely used and could even be called *de facto* standards. Our method of converting from *PV* to SIP distortion is explained in §3. We describe the software we developed to bridge the gap between distortion representations and share our PTF experience gained in §4. The software's performance is reported in §5. We discuss our findings in §6, and give our recommendations for tool developers and concluding remarks in §7 and §8, respectively.

2. THE *PV* AND SIP DISTORTION REPRESENTATIONS

The *PV* distortion representation is implemented in SCAMP, SWarp, and SExtractor. It is performed in intermediate world coordinate space.² The intermediate longitudinal and latitudinal corrections are seventh-order polynomials, and their coefficients are FITS-header parameters *PVi-j*, where $i = 1, 2$ are axis indices, and $j = 0, \dots, 39$ specify indices to the numeric values for the terms in the polynomials. The polynomials are functions of x and y , the uncorrected longitudinal and latitudinal offsets from the distortion-center origin, respectively, and $r = \sqrt{x^2 + y^2}$. Note that while these tools read the radial terms, SCAMP does not compute any radial terms, and omits the corresponding *PVi-j* keywords from its output. The full equations for the corrections, taken directly from the SExtractor source code, are given as follows:

*See <http://astromatic.net/forum/attachment.php?aid=220> for an excerpt of this early draft.

[†]Aperture Photometry Tool, which was developed by one of us (R.R.L.), features a very nice FITS-image viewer that incorporates either seventh-order *PV* or ninth-order SIP distortion in its astrometric calculations (download from www.aperturephotometry.org).

$$\begin{aligned}
x' = & \quad PV1.0 + PV1.1x + PV1.2y + PV1.3r + PV1.4x^2 + PV1.5xy + PV1.6y^2 + PV1.7x^3 + \\
& \quad PV1.8x^2y + PV1.9xy^2 + PV1.10y^3 + PV1.11r^3 + PV1.12x^4 + PV1.13x^3y + PV1.14x^2y^2 \\
& \quad PV1.15xy^3 + PV1.16y^4 + PV1.17x^5 + PV1.18x^4y + PV1.19x^3y^2 + PV1.20x^2y^3 + PV1.21xy^4 \\
& \quad PV1.22y^5 + PV1.23r^5 + PV1.24x^6 + PV1.25x^5y + PV1.26x^4y^2 + PV1.27x^3y^3 + PV1.28x^2y^4 \\
& \quad PV1.29xy^5 + PV1.30y^6 + PV1.31x^7 + PV1.32x^6y + PV1.33x^5y^2 + PV1.34x^4y^3 + PV1.35x^3y^4 \\
& \quad \quad PV1.36x^2y^5 + PV1.37xy^6 + PV1.38y^7 + PV1.39r^7
\end{aligned} \tag{1}$$

$$\begin{aligned}
y' = & \quad PV2.0 + PV2.1y + PV2.2x + PV2.3r + PV2.4y^2 + PV2.5xy + PV2.6x^2 + PV2.7y^3 + \\
& \quad PV2.8xy^2 + PV2.9x^2y + PV2.10x^3 + PV2.11r^3 + PV2.12y^4 + PV2.13xy^3 + PV2.14x^2y^2 \\
& \quad PV2.15x^3y + PV2.16x^4 + PV2.17y^5 + PV2.18xy^4 + PV2.19x^2y^3 + PV2.20x^3y^2 + PV2.21x^4y \\
& \quad PV2.22x^5 + PV2.23r^5 + PV2.24y^6 + PV2.25xy^5 + PV2.26x^2y^4 + PV2.27x^3y^3 + PV2.28x^4y^2 \\
& \quad PV2.29x^5y + PV2.30x^6 + PV2.31y^7 + PV2.32xy^6 + PV2.33x^2y^5 + PV2.34x^3y^4 + PV2.35x^4y^3 \\
& \quad \quad PV2.36x^5y^2 + PV2.37x^6y + PV2.38x^7 + PV2.39r^7
\end{aligned} \tag{2}$$

where x' is the distortion-corrected intermediate longitude, and y' is the distortion-corrected intermediate latitude. Eqs. 1 and 2 have nearly identical forms (by interchanging x and y), and are identical to those in the proposed TPV distortion representation.[‡] The keyword numbering does not indicate the order of the corresponding polynomial term. As can be seen, the corrections have constant and linear terms, which must be dealt with specially when converting PV distortion into SIP distortion. In our experience, SCAMP does not compute the radial terms, and the polynomial coefficients associated with them are not written to the output FITS header. Indeed, only coefficients with non-zero values are outputted. For the undistorted case, $PV1.1 = PV2.1 = 1$ and all other coefficients are zero; this leads to the trivial case of $x' = x$ and $y' = y$. For light and moderate distortion, $PV1.1$ and $PV2.1$ have values near unity, and the remaining coefficients have values close to zero.

Currently, SCAMP, SWarp, and SExtractor do not expect a special setting of the `CTYPEn` FITS keywords (in versions 1.7.0, 2.19.1, and 2.8.6, respectively). These software programs simply check whether the PVi_j keywords are present in the FITS header, and, if so, they apply the PV distortion correction.

The SIP distortion convention has been documented elsewhere⁸ and is briefly summarized here. We define u and v as relative pixel coordinates with origin at `CRPIX1`, `CRPIX2`. As above, x' and y' are “intermediate world coordinates” in degrees, with origin at `CRVAL1`, `CRVAL2`.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} CD1.1 & CD1.2 \\ CD2.1 & CD2.2 \end{pmatrix} \begin{pmatrix} u + \sum_{i,j} A_{i,j} u^i v^j \\ v + \sum_{i,j} B_{i,j} u^i v^j \end{pmatrix} \tag{3}$$

where the sum of i and j is allowed to range from 2 to 9, with the maximum specified by FITS keywords `A_ORDER` and `B_ORDER`, and the values of $A_{i,j}$ and $B_{i,j}$ are represented in the FITS header as `A_i_j` and `B_i_j`.

Since the $CD_{i,j}$ keywords encode skew as well as rotation and scaling, it is strongly discouraged to use constant and linear terms in the SIP polynomials. The CD -matrix values together with the higher-order distortion polynomials, as in Equation 3, define a unique transformation from pixel coordinates to the plane-of-projection.

For Spitzer, SIP also provides polynomials for the reverse transformation, for fast inversion. Corrected pixel coordinates U and V are found from

$$\begin{pmatrix} U \\ V \end{pmatrix} = CD^{-1} \begin{pmatrix} x' \\ y' \end{pmatrix}, \tag{4}$$

[‡]<http://fits.gsfc.nasa.gov/registry/tpvwcs/tpv.html>

and then the original pixel coordinates are computed using the reverse SIP coefficients by

$$u = U + \sum_{p,q} \text{AP}_{-p-q} U^p V^q, \quad p + q \leq \text{AP_ORDER}, \quad (5)$$

$$v = V + \sum_{p,q} \text{BP}_{-p-q} U^p V^q, \quad p + q \leq \text{BP_ORDER}. \quad (6)$$

Here, then, are the chief differences between *PV* and SIP. The SIP convention makes distortion corrections to the image pixel coordinates, whereas *PV* distortion involves corrections to sky coordinates. The SIP convention encourages only quadratic and higher polynomial terms in pixel axes u and v . *PV* distortion has polynomial terms in x , y , and radius from the distortion center. Unlike *PV* distortion, the SIP convention provides for two different sets of FITS keywords: one for conversion from pixel coordinates to sky coordinates, and the other for the reverse transformation. (We will return to the issue of whether the reverse coefficients are really necessary in Section 6.) The SIP convention modifies the `CTYPE1` and `CTYPE2` values by appending “-SIP” to their settings.

For the case of what SCAMP actually does, that is, to omit the radial terms, it can be seen from the equations in this section that an exact transformation between SIP and *PV* polynomials is possible, by multiplying out by the CD matrix.

3. METHOD OF *PV* TO SIP CONVERSION

Fig. 1 gives a flow chart of our method for conversion from *PV* distortion to SIP distortion. Because SCAMP omits the radial terms in Eqs. 1 and 2, one can solve algebraically for the exact relationship between *PV* and SIP polynomials for the forward transformation. All algebra was all done with Sage using symbolic calculations.[§] One of the features of the method is that we follow good form by folding the constant *PV* terms (`PV1.0` and `PV2.0`) into the `CRVAL1` and `CRVAL2` values, and the linear terms (`PV1.1`, `PV1.2`, `PV2.1`, and `PV2.2`)¶ into a recomputed CD matrix. The method also recomputes the `PVi_j` keywords to match this new CD matrix, constraining `PV1.1` = `PV2.1` = 1 and `PV1.0` = `PV1.2` = `PV2.0` = `PV2.2` = 0, so that only the quadratic and higher-order *PV* terms are needed. The resulting *PV* and SIP distortion polynomials then properly contain only the quadratic and higher-order terms. Consequently, the new sets of *PV* and SIP coefficients can coexist in the same FITS header. This coexistence is allowed because the new CD matrix can be shared between the two distortion representations, and most software packages that handle distortion currently will apply only one representation and ignore the other.

Appendix A gives the equations for computing the new CD matrix, the SIP coefficients, and the corrections to `CRVAL1` & `CRVAL2`.

4. SOFTWARE

Based on the method described in §3, we developed two different software modules to achieve flexibility in representing geometric distortion in FITS images. The `pv2sip` module converts `PVi_j` distortion keywords into SIP distortion keywords. The `sip2pv` module performs the opposite conversion. Both modules are written in the C language and optimized for the fastest processing speed. A slower version of the `pv2sip` module was initially developed in Python as a prototype. The modules handle *PV* distortion up to the fourth polynomial order (cf. the limit of SExtractor is seventh order), and SIP distortion up to the fourth polynomial order (cf. the limit of the *WCSTools* library, underlying *DS9*, is ninth order). FITS-file I/O is handled by the CFITSIO library. The modules have been tested on both Linux and Mac machines, and have been benchmarked at 3.7 s (`pv2sip`) and 0.5 s (`sip2pv`) for respective conversion of a single PTF image on a MacBook Pro laptop with a 2.2 GHz Intel Core i7 processor and 4 GB of memory and running Lion OS X.

The PTF pipelines routinely run the `pv2sip` module after SCAMP finds the astrometric solution to fourth polynomial order. If SCAMP fails to find a solution, then a similar Astrometry.net run is attempted, and if

[§]<http://www.sagemath.org/doc/reference/index.html>

[¶]Technically, the distortion part of the `PV1.1` and `PV2.1` terms is the deviation from unity.

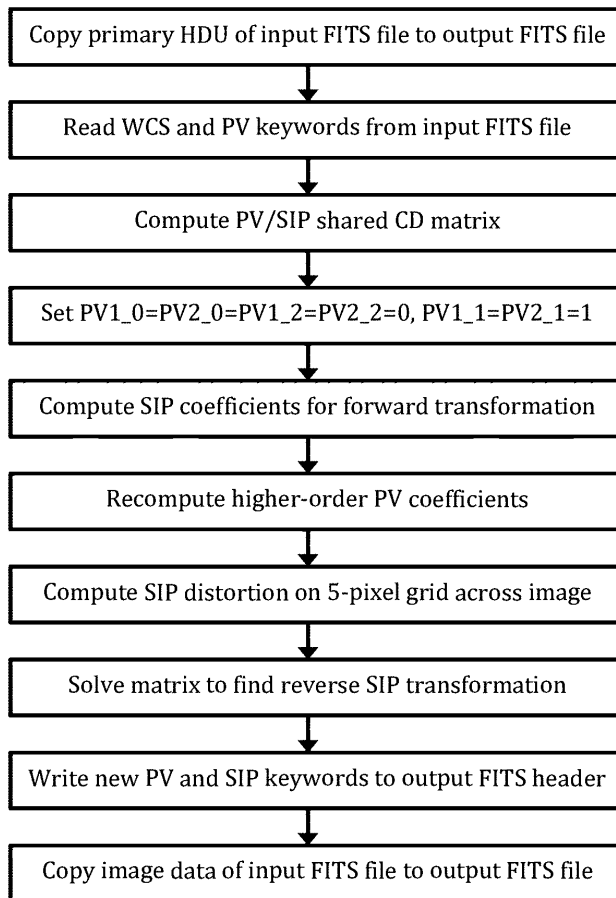


Figure 1. Flow chart of method of *PV* to SIP conversion.

it succeeds, then the *sip2pv* module is executed. The resulting FITS header will simultaneously contain the two different representations of geometric distortion, which can coexist in the FITS header, as explained in §3. Regardless of the software that performed the astrometric calibration, SExtractor is subsequently executed to create a source catalog from the image. The “-SIP” string appended to the CTYPE1 and CTYPE2 keywords are ignored by SExtractor, and the *PV* distortion is included in the computation of the catalog’s equatorial coordinates. The catalog will therefore have astrometry with the expected accuracy. As an example, Appendix B lists the FITS keywords in a PTF-image header that are relevant to the WCS solution and both geometric-distortion representations. A FITS viewer that handles SIP distortion, such as *DS9* or Aperture Photometry Tool,⁹ will also compute accurate astrometry for the image, as expected.

5. PERFORMANCE

The software has been executed in our pipelines on more than 2 million PTF images. The *pv2sip* module has proven itself to be quite robust, with a failure rate of only $\approx 0.0116\%$. Through Perl multithreading in the pipelines, seven instances of the *pv2sip* module are executed simultaneously on a CPU with 8 cores running Linux, and 1.39 s per instance is achieved.

With the availability of our *PV*-to-SIP and SIP-to-*PV* converters, we can now settle the question of whether the reverse coefficients are strictly necessary. In tests with the Aperture Photometry Tool on a PTF image of converting a source list in celestial coordinates to pixel coordinates, the speed of the inverting the *PV* coefficients was nearly the same (within a percent) as using the pre-computed SIP reverse polynomials. Results with other implementations will likely vary, but this result does indicate that a policy of providing the coefficients for the

pixel-to-sky conversion only is sound, at least for the fourth-order distortion in PTF images. However, it is generally more robust to compute the pixel-to-sky coordinates from reverse polynomial coefficients (such as are provided by the SIP convention) than to rely on iterative inversion of the distortion polynomial.

6. DISCUSSION

For users of processed data products from many observatories, the distortion representation does not matter much. This is because the observatory or project provides software to re-project the distorted images onto a uniform celestial grid. A prime example is the MOPEX package developed by the SSC (Ref. 10). A user of the “finished” Spitzer products from MOPEX will be using its distortionless maps or mosaics, which contain only the standard WCS FITS keywords defined in Ref. 3. Even for users wanting to go back to the “Basic Calibrated Data” Spitzer images, if they use the MOPEX package as recommended to make the final maps for their scientific analyses, they will not have to deal directly with distortion.

For PTF, it is critical to preserve the individual images without remapping onto a sky grid, to best preserve the ability to detect and measure transient events. Users of PTF images, therefore, need access to a wider range of tools than has previously been the case.

The conversion tools discussed in this manuscript allow use of a wide variety of tools, and especially those that have supported the SIP convention for nearly a decade, with the confidence that these tools have been verified to work properly, owing to the many images produced and still being produced by Spitzer. These tools include:

- *MOPEX* from the SSC (Ref. 10)
- IPAC’s *Skyview* (<http://www.ipac.caltech.edu/Skyview>)
- *WCSTools*, which underlies *SAOImage* and *DS9* image viewers (<http://tdc-www.harvard.edu/wcstools>)
- *StarLink*, which underlies GAIA. (<http://starlink.jach.hawaii.edu/starlink/AST>)
- *Drizzle* from Hook & Fruchter (Ref. 11)
- IDL Astronomy User’s Library (<http://idlastro.gsfc.nasa.gov>)
- Aperture Photometry Tool^{||} from Laher *et al.* (Ref. 9)
- *Montage*, the grid-capable mosaicker available from IPAC (<http://montage.ipac.caltech.edu>)
- ICORE** from Masci & Fowler (Ref. 12)
- *Pywcs*, the STScI WCS Python package (<http://stsdas.stsci.edu/astrolib/pywcs/>)

Since we have verified that these tools work in SIP-mode with both *PV* and SIP FITS keywords present, we have programmed our conversion software to leave both in. (Note: we will probably upgrade the software to have an option to control whether to omit or retain the other scheme’s keywords after conversion.)

Recently, the NOAO Dark Energy Consortium has moved to formalize the *PV* scheme implemented in SCAMP and SExtractor into a “TPV” convention. As of this writing, the comment period is open on the FITS Convention Registry.^{††} The differences between the TPV scheme and what we have described for *PV* are that TPV applies only to the standard gnomonic projection, and TPV changes the *CTYPE1* and *CTYPE2* keywords for the standard gnomonic projection to end in “TPV” instead of “TAN”. Thus, the TPV convention *only*

^{||}It bears repeating that Aperture Photometry Tool has a FITS-image viewer that includes either seventh-order *PV* or ninth-order SIP distortion in its astrometric calculations, depending on the contents in the FITS header (download from www.aperturephotometry.org).

**<http://wise2.ipac.caltech.edu/staff/fmasci/awaicpub.html>

††http://fits.gsfc.nasa.gov/fits_registry.html

builds on the “TAN” projection, and *nonobvious* *CTYPEn* modifications would be needed for compatibility with other standard projections. The current TPV proposal makes the *CTYPE1* and *CTYPE2* settings the “trigger” for software to look for and apply the *PV* coefficients, whereas no special trigger is needed at all for the current *PV* implementation.

For SIP, the “trigger” is the addition of “-SIP” to the existing *CTYPE1* and *CTYPE2* values. This was proposed during the now-defunct effort to define “Paper IV” on distortion as part of the FITS standard. Regrettably, this fine idea was later discarded and the FITS standard now states that *CTYPEn* should be eight characters only. The advantage of the 12-character *CTYPEn* of the SIP convention gives it the ability to work not only with gnomonic projections, but with any of the standard projections in Paper II (Ref. 3).

Another key difference between *PV*/TPV and SIP is that SIP has the option to include reverse coefficients. Our converter will estimate these as part of the *PV*-to-SIP process. The historical reason for the inclusion of reverse coefficients in SIP once again stems from Spitzer. The MOPEX mosaicker mentioned earlier operates by mapping the pixel corners of the undistorted celestial map grid (i.e., the output grid) onto the distorted image. This process is repeated for every pixel of every single IRAC, MIPS, and IRS-peakup image, every time the Spitzer Science Center reprocesses the data. Therefore, it was decided to include the reverse coefficients in the Spitzer headers to avoid a potentially slow iterative solution for the sky-to-pixels direction.

The *PV* method of computing the distortion coefficients, as implemented in SCAMP, SWarp, and SExtractor and adopted by PTF, means that only polynomials in x and y (or historically in ξ and η) are used. The radial terms are not used. The TPV sample header *also* does not use the radial terms.^{‡‡} Therefore, there is *good reason* for a user to adopt the scheme that PTF has demonstrated to work already. Indeed, it removes the need for the TPV representation entirely. SWarp and SExtractor, which compute the *PV* rather than the SIP distortion correction, ignore the “-SIP” suffix in the *CTYPEn* keywords, as well as the SIP keywords (if present), and instead read in and apply the *PVi-j* keywords if they are present in the image’s FITS header. In the absence of the “-SIP” suffix and SIP keywords, other software tools can be modified to have this very reasonable behavior, too, as indeed has already been done for Aperture Photometry Tool.

7. RECOMMENDATIONS FOR TOOL DEVELOPMENT

Developers of tools for handling astronomical data that read the distortion keywords for use in coordinate conversions typically want to support not only what has been formally ratified as a standard, but also those conventions that appear in the data products that astronomers use. Here are our recommendations for tool developers:

- For software packages that *produce* the distortion keywords, implement either the SIP convention or the *PV* representation.
- For tools that *consume* the FITS WCS and distortion information, in order to compute astrometry for image pixels, use either of the following logic branches, depending on the *CTYPE1* and *CTYPE2* keyword settings:
 - If the *CTYPEn* setting ends with “-SIP”, then apply the SIP coefficients.
 - Else, if the *CTYPEn* setting is a FITS-standard, such as ‘RA—TAN’ and ‘DEC—TAN’, then apply the *PV* coefficients if they are present in the FITS header.

For software or large projects that *compute* distortion keywords, we recommend the following:

- Include both SIP and *PV* coefficients in the FITS headers of achived image products, if desirable and warranted, or leave it to the users to run the *PV*-to-SIP and SIP-to-*PV* converters as needed.
- If using the SIP convention, be sure to supply both the forward and reverse coefficients, as many software packages have implementations for both sets of polynomials.

^{‡‡}<http://fits.gsfc.nasa.gov/registry/tpvwcs/tpv-hdr.txt>

- Follow good form by folding any constant and linear distortion terms into the CD matrix, and leave only the quadratic and higher-order terms to the distortion convention that you use.

For users of data that include distortion polynomials, we suggest considering these points:

- If you will be using the distorted data to construct an undistorted final map, just make sure that your map-making software will properly handle the distortion polynomials. Afterwards, you will need not worry further about the details of the distortion representation.
- If you will be using the distorted data in other tools, consider running the *PV*-to-SIP or SIP-to-*PV* converter to ensure that your data will be compatible with the widest range of tools.
- Test the distorted data with the tools you are using to ensure the coordinate conversions are correct and consistent.
- Software development is always an ongoing effort, and it is not inconceivable that, for the actively maintained software that does not currently support the desired distortion representation, there will be in the future an upgrade for the capability, especially if an interest for that capability is expressed to the developer. You might need to ask for it, and you may be surprised to actually receive it.

8. CONCLUSION

Geometrical distortion in astronomical images and the software tools that are distortion-capable have been the subject of this paper. The method and software developed in this effort for converting *PV* distortion into SIP distortion and vice versa, along with their utilization in the PTF image-processing pipelines, have been described. The benefit is relevant WCS keywords in FITS image headers for both *PV* and SIP distortion representations, and this allows both the generation of SExtractor catalogs with accurate astrometry and a wider variety of FITS viewers and tools to compute astrometry with distortion corrections for more accurate analysis. Finally, we have also made a strong case for why the proposed new TPV distortion representation is superfluous. Its CTYPE*n* settings step on “TAN” in a way that is incompatible with the inclusion of other standard projections (which require more than one character to identify them unambiguously). TPV also destroys the mutual coexistence of SIP and *PV* that is currently possible.

Our conversion software developed to date and relevant PTF experience have demonstrated the proof of concept. Future work to upgrade the software from fourth to seventh polynomial order is planned.

APPENDIX A. EQUATIONS FOR *PV* TO SIP CONVERSION

This section includes the equations for converting fourth-order *PV*_{*i,j*} keywords, as computed by SCAMP in the PTF pipeline, into the SIP representation. As noted in §3, in converting from the *PV* representation to the SIP convention, we follow good form by folding all the constant and linear terms into the FITS standard keywords. The “primed” keywords are the new ones.

$$\begin{pmatrix} U_{\text{diff}} \\ V_{\text{diff}} \end{pmatrix} = CD^{-1} \begin{pmatrix} PV1.0 \\ PV2.0 \end{pmatrix}$$

$$CD'_{11} = CD_{11} \cdot PV1.1 + CD_{21} \cdot PV1.2$$

$$CD'_{12} = CD_{12} \cdot PV1.1 + CD_{22} \cdot PV1.2$$

$$CD'_{21} = CD_{11} \cdot PV2.2 + CD_{21} \cdot PV2.1$$

$$CD'_{22} = CD_{12} \cdot PV2.2 + CD_{22} \cdot PV2.1$$

$$\begin{pmatrix} x_{\text{diff}} \\ y_{\text{diff}} \end{pmatrix} = CD' \begin{pmatrix} U_{\text{diff}} \\ V_{\text{diff}} \end{pmatrix}$$

$$CRVAL1' = CRVAL1 + x_{\text{diff}} / \cos(CRVAL2 \cdot \pi / 180)$$

$$CRVAL2' = CRVAL2 + y_{\text{diff}}$$

Define the intermediate polynomials $k_{i,j}$ and $l_{i,j}$ as follows:

$$k_{0,2} = CD_{12}^2 \cdot PV1.4 + CD_{12} \cdot CD_{22} \cdot PV1.5 + CD_{22}^2 \cdot PV1.6$$

$$l_{0,2} = CD_{12}^2 \cdot PV2.6 + CD_{12} \cdot CD_{22} \cdot PV2.5 + CD_{22}^2 \cdot PV2.4$$

$$k_{1,1} = 2 \cdot CD_{11} \cdot CD_{12} \cdot PV1.4 + CD_{11} \cdot CD_{22} \cdot PV1.5 + CD_{12} \cdot CD_{21} \cdot PV1.5 \\ + 2 \cdot CD_{21} \cdot CD_{22} \cdot PV1.6$$

$$l_{1,1} = 2 \cdot CD_{11} \cdot CD_{12} \cdot PV2.6 + CD_{11} \cdot CD_{22} \cdot PV2.5 + CD_{12} \cdot CD_{21} \cdot PV2.5 \\ + 2 \cdot CD_{21} \cdot CD_{22} \cdot PV2.4$$

$$k_{0,2} = CD_{12}^2 \cdot PV1.4 + CD_{12} \cdot CD_{22} \cdot PV1.5 + CD_{22}^2 \cdot PV1.6$$

$$l_{0,2} = CD_{12}^2 \cdot PV2.6 + CD_{12} \cdot CD_{22} \cdot PV2.5 + CD_{22}^2 \cdot PV2.4$$

$$k_{0,3} = CD_{12}^3 \cdot PV1.7 + CD_{12}^2 \cdot CD_{22} \cdot PV1.8 + CD_{12} \cdot CD_{22}^2 \cdot PV1.9 + CD_{22}^3 \cdot PV1.10$$

$$l_{0,3} = CD_{12}^3 \cdot PV2.10 + CD_{12}^2 \cdot CD_{22} \cdot PV2.9 + CD_{12} \cdot CD_{22}^2 \cdot PV2.8 + CD_{22}^3 \cdot PV2.7$$

$$k_{1,2} = 3 \cdot CD_{11} \cdot CD_{12}^2 \cdot PV1.7 + 2 \cdot CD_{11} \cdot CD_{12} \cdot CD_{22} \cdot PV1.8 + CD_{11} \cdot CD_{22}^2 \cdot PV1.9 \\ + CD_{12}^2 \cdot CD_{21} \cdot PV1.8 + 2 \cdot CD_{12} \cdot CD_{21} \cdot CD_{22} \cdot PV1.9 + 3 \cdot CD_{21} \cdot CD_{22}^2 \cdot PV1.10$$

$$l_{1,2} = 3 \cdot CD_{11} \cdot CD_{12}^2 \cdot PV2.10 + 2 \cdot CD_{11} \cdot CD_{12} \cdot CD_{22} \cdot PV2.9 + CD_{11} \cdot CD_{22}^2 \cdot PV2.8 \\ + CD_{12}^2 \cdot CD_{21} \cdot PV2.9 + 2 \cdot CD_{12} \cdot CD_{21} \cdot CD_{22} \cdot PV2.8 + 3 \cdot CD_{21} \cdot CD_{22}^2 \cdot PV2.7$$

$$k_{2,1} = 3 \cdot CD_{11}^2 \cdot CD_{12} \cdot PV1.7 + CD_{11}^2 \cdot CD_{22} \cdot PV1.8 + 2 \cdot CD_{11} \cdot CD_{12} \cdot CD_{21} \cdot PV1.8 \\ + 2 \cdot CD_{11} \cdot CD_{21} \cdot CD_{22} \cdot PV1.9 + CD_{12} \cdot CD_{21}^2 \cdot PV1.9 + 3 \cdot CD_{21}^2 \cdot CD_{22} \cdot PV1.10$$

$$l_{2,1} = 3 \cdot CD_{11}^2 \cdot CD_{12} \cdot PV2.10 + CD_{11}^2 \cdot CD_{22} \cdot PV2.9 + 2 \cdot CD_{11} \cdot CD_{12} \cdot CD_{21} \cdot PV2.9 \\ + 2 \cdot CD_{11} \cdot CD_{21} \cdot CD_{22} \cdot PV2.8 + CD_{12} \cdot CD_{21}^2 \cdot PV2.8 + 3 \cdot CD_{21}^2 \cdot CD_{22} \cdot PV2.7$$

$$k_{3,0} = CD_{11}^3 \cdot PV1.7 + CD_{11}^2 \cdot CD_{21} \cdot PV1.8 + CD_{11} \cdot CD_{21}^2 \cdot PV1.9 + CD_{21}^3 \cdot PV1.10$$

$$l_{3,0} = CD_{11}^3 \cdot PV2.10 + CD_{11}^2 \cdot CD_{21} \cdot PV2.9 + CD_{11} \cdot CD_{21}^2 \cdot PV2.8 + CD_{21}^3 \cdot PV2.7$$

$$k_{0,4} = CD_{12}^4 \cdot PV1.12 + CD_{12}^3 \cdot CD_{22} \cdot PV1.13 + CD_{12}^2 \cdot CD_{22}^2 \cdot PV1.14 \\ + CD_{12} \cdot CD_{22}^3 \cdot PV1.15 + CD_{22}^4 \cdot PV1.16$$

$$l_{0,4} = CD_{12}^4 \cdot PV2.16 + CD_{12}^3 \cdot CD_{22} \cdot PV2.15 + CD_{12}^2 \cdot CD_{22}^2 \cdot PV2.14 \\ + CD_{12} \cdot CD_{22}^3 \cdot PV2.13 + CD_{22}^4 \cdot PV2.12$$

$$k_{1,3} = 4 \cdot CD_{11} \cdot CD_{12}^3 \cdot PV1.12 + 3 \cdot CD_{11} \cdot CD_{12}^2 \cdot CD_{22} \cdot PV1.13 + 2 \cdot CD_{11} \cdot CD_{12} \cdot CD_{22}^2 \cdot PV1.14 \\ + CD_{11} \cdot CD_{22}^3 \cdot PV1.15 + CD_{12}^3 \cdot CD_{21} \cdot PV1.13 + 2 \cdot CD_{12}^2 \cdot CD_{21} \cdot CD_{22} \cdot PV1.14 \\ + 3 \cdot CD_{12} \cdot CD_{21} \cdot CD_{22}^2 \cdot PV1.15 + 4 \cdot CD_{21} \cdot CD_{22}^3 \cdot PV1.16$$

$$l_{1,3} = 4 \cdot CD_{11} \cdot CD_{12}^3 \cdot PV2.16 + 3 \cdot CD_{11} \cdot CD_{12}^2 \cdot CD_{22} \cdot PV2.15 + 2 \cdot CD_{11} \cdot CD_{12} \cdot CD_{22}^2 \cdot PV2.14 \\ + CD_{11} \cdot CD_{22}^3 \cdot PV2.13 + CD_{12}^3 \cdot CD_{21} \cdot PV2.15 + 2 \cdot CD_{12}^2 \cdot CD_{21} \cdot CD_{22} \cdot PV2.14 \\ + 3 \cdot CD_{12} \cdot CD_{21} \cdot CD_{22}^2 \cdot PV2.13 + 4 \cdot CD_{21} \cdot CD_{22}^3 \cdot PV2.12$$

$$k_{2,2} = 6 \cdot CD_{11}^2 \cdot CD_{12}^2 \cdot PV1.12 + 3 \cdot CD_{11}^2 \cdot CD_{12} \cdot CD_{22} \cdot PV1.13 + CD_{11}^2 \cdot CD_{22}^2 \cdot PV1.14 \\ + 3 \cdot CD_{11} \cdot CD_{12}^2 \cdot CD_{21} \cdot PV1.13 + 4 \cdot CD_{11} \cdot CD_{12} \cdot CD_{21} \cdot CD_{22} \cdot PV1.14$$

$$\begin{aligned}
& + 3 \cdot CD_{11} \cdot CD_{21} \cdot CD_{22}^2 \cdot PV1.15 + CD_{12}^2 \cdot CD_{21}^2 \cdot PV1.14 \\
& + 3 \cdot CD_{12} \cdot CD_{21}^2 \cdot CD_{22} \cdot PV1.15 + 6 \cdot CD_{21}^2 \cdot CD_{22}^2 \cdot PV1.16 \\
l_{2,2} = & 6 \cdot CD_{11}^2 \cdot CD_{12}^2 \cdot PV2.16 + 3 \cdot CD_{11}^2 \cdot CD_{12} \cdot CD_{22} \cdot PV2.15 + CD_{11}^2 \cdot CD_{22}^2 \cdot PV2.14 \\
& + 3 \cdot CD_{11} \cdot CD_{12}^2 \cdot CD_{21} \cdot PV2.15 + 4 \cdot CD_{11} \cdot CD_{12} \cdot CD_{21} \cdot CD_{22} \cdot PV2.14 + \\
& 3 \cdot CD_{11} \cdot CD_{21} \cdot CD_{22}^2 \cdot PV2.13 + CD_{12}^2 \cdot CD_{21}^2 \cdot PV2.14 \\
& + 3 \cdot CD_{12} \cdot CD_{21}^2 \cdot CD_{22} \cdot PV2.13 + 6 \cdot CD_{21}^2 \cdot CD_{22}^2 \cdot PV2.12 \\
k_{3,1} = & 4 \cdot CD_{11}^3 \cdot CD_{12} \cdot PV1.12 + CD_{11}^3 \cdot CD_{22} \cdot PV1.13 + 3 \cdot CD_{11}^2 \cdot CD_{12} \cdot CD_{21} \cdot PV1.13 \\
& + 2 \cdot CD_{11}^2 \cdot CD_{21} \cdot CD_{22} \cdot PV1.14 + 2 \cdot CD_{11} \cdot CD_{12} \cdot CD_{21}^2 \cdot PV1.14 \\
& + 3 \cdot CD_{11} \cdot CD_{21}^2 \cdot CD_{22} \cdot PV1.15 + CD_{12} \cdot CD_{21}^3 \cdot PV1.15 + 4 \cdot CD_{21}^3 \cdot CD_{22} \cdot PV1.16 \\
l_{3,1} = & 4 \cdot CD_{11}^3 \cdot CD_{12} \cdot PV2.16 + CD_{11}^3 \cdot CD_{22} \cdot PV2.15 + 3 \cdot CD_{11}^2 \cdot CD_{12} \cdot CD_{21} \cdot PV2.15 \\
& + 2 \cdot CD_{11}^2 \cdot CD_{21} \cdot CD_{22} \cdot PV2.14 + 2 \cdot CD_{11} \cdot CD_{12} \cdot CD_{21}^2 \cdot PV2.14 \\
& + 3 \cdot CD_{11} \cdot CD_{21}^2 \cdot CD_{22} \cdot PV2.13 + CD_{12} \cdot CD_{21}^3 \cdot PV2.13 + 4 \cdot CD_{21}^3 \cdot CD_{22} \cdot PV2.12 \\
k_{4,0} = & CD_{11}^4 \cdot PV1.12 + CD_{11}^3 \cdot CD_{21} \cdot PV1.13 + CD_{11}^2 \cdot CD_{21}^2 \cdot PV1.14 \\
& + CD_{11} \cdot CD_{21}^3 \cdot PV1.15 + CD_{21}^4 \cdot PV1.16 \\
l_{4,0} = & CD_{11}^4 \cdot PV2.16 + CD_{11}^3 \cdot CD_{21} \cdot PV2.15 + CD_{11}^2 \cdot CD_{21}^2 \cdot PV2.14 \\
& + CD_{11} \cdot CD_{21}^3 \cdot PV2.13 + CD_{21}^4 \cdot PV2.12
\end{aligned}$$

Note that much of the complication in these equations is caused by the arbitrary indices assigned to the PV coefficients. The next step is to compute the inverse of the CD' matrix as CD'^{-1} . Then, the SIP coefficients are computed as follows:

$$\begin{pmatrix} A_{i,j} \\ B_{i,j} \end{pmatrix} = CD'^{-1} \begin{pmatrix} k_{i,j} \\ l_{i,j} \end{pmatrix}.$$

The reverse-SIP coefficients are calculated and the PV coefficients are recalculated by our software, but these equations are not shown here.

APPENDIX B. SAMPLE FITS HEADER

For a sample PTF image (with processed-image ID = 13051948), below is a listing of the portion of its FITS header that is relevant to the WCS solution and the two representations of geometric distortion contained therein. See §4 for a discussion of this example. Notice the missing SCAMP-generated PV_{i-j} keywords that correspond to the radial terms in Eqs. 1 and 2.

```

/ ASTROMETRY

CRVAL1 =      104.758177886399 / [deg] RA of reference point
CRVAL2 =      17.5110457095458 / [deg] DEC of reference point
CRPIX1 =              767.6599731
CRPIX2 =              1732.279053
CTYPE1 = 'RA---TAN-SIP'      / TAN (gnomic) projection + SIP distortions
CTYPE2 = 'DEC--TAN-SIP'      / TAN (gnomic) projection + SIP distortions
CUNIT1 = 'deg      '         / Image axis-1 celestial-coordinate units
CUNIT2 = 'deg      '         / Image axis-2 celestial-coordinate units
CRTYPE1 = 'deg      '         / Data units of CRVAL1
CRTYPE2 = 'deg      '         / Data units of CRVAL2
CD1_1   = 0.000281189660249318 / Transformation matrix

```

CD1_2 = 1.72818762145632E-06
CD2_1 = 1.90270663006752E-06
CD2_2 = -0.000281108762529357
PIXSCALE= 1.01 / [arcsec/pix] Pixel scale
WCSAXES = 2
EQUINOX = 2000. / [yr] Equatorial coordinates definition
LONPOLE = 180.
LATPOLE = 0.

/ SCAMP DISTORTION KEYWORDS

RADECSYS= 'ICRS' / Astrometric system
PV1_0 = 0. / Projection distortion parameter
PV1_1 = 1. / Projection distortion parameter
PV1_2 = 0. / Projection distortion parameter
PV1_4 = 0.000404939572314492 / Projection distortion parameter
PV1_5 = -0.000697944221451579 / Projection distortion parameter
PV1_6 = 0.000169919117675088 / Projection distortion parameter
PV1_7 = -0.000525380713396686 / Projection distortion parameter
PV1_8 = 0.00026872477049526 / Projection distortion parameter
PV1_9 = -0.000294326113855124 / Projection distortion parameter
PV1_10 = -0.000140354952126735 / Projection distortion parameter
PV1_12 = 0.00193906630022206 / Projection distortion parameter
PV1_13 = -0.000700555630422411 / Projection distortion parameter
PV1_14 = -0.00019841420224509 / Projection distortion parameter
PV1_15 = 0.000120125448252655 / Projection distortion parameter
PV1_16 = -0.000204494383079758 / Projection distortion parameter
PV2_0 = 0. / Projection distortion parameter
PV2_1 = 1. / Projection distortion parameter
PV2_2 = 0. / Projection distortion parameter
PV2_4 = -0.00082046810054367 / Projection distortion parameter
PV2_5 = 0.000449503620677844 / Projection distortion parameter
PV2_6 = -8.30093146297761E-05 / Projection distortion parameter
PV2_7 = -0.000175962454767749 / Projection distortion parameter
PV2_8 = 0.000216392042755894 / Projection distortion parameter
PV2_9 = -0.000542596339918239 / Projection distortion parameter
PV2_10 = 0.000198643540669774 / Projection distortion parameter
PV2_12 = 8.37877112535033E-05 / Projection distortion parameter
PV2_13 = -0.000310384155458062 / Projection distortion parameter
PV2_14 = -0.000642802898226449 / Projection distortion parameter
PV2_15 = 0.000873959922357863 / Projection distortion parameter
PV2_16 = -0.0011573088289513 / Projection distortion parameter

/ SIP DISTORTION KEYWORDS

A_ORDER = 4 / Distortion order for A
A_0_2 = 4.7537871143152E-08 / Projection distortion parameter
A_0_3 = 1.10298569625795E-11 / Projection distortion parameter
A_0_4 = -4.54597110704134E-15 / Projection distortion parameter
A_1_1 = 1.96175741473304E-07 / Projection distortion parameter
A_1_2 = -2.36396755175757E-11 / Projection distortion parameter
A_1_3 = -2.55645920974982E-15 / Projection distortion parameter

A_2_0 = 1.12396045844352E-07 / Projection distortion parameter
A_2_1 = -2.14241926496027E-11 / Projection distortion parameter
A_2_2 = -4.14899903101719E-15 / Projection distortion parameter
A_3_0 = -4.13014041971452E-11 / Projection distortion parameter
A_3_1 = 1.65654022221544E-14 / Projection distortion parameter
A_4_0 = 4.2846501292706E-14 / Projection distortion parameter
A_DMAX = 0.795029513863436 / Projection distortion parameter
B_ORDER = 4 / Distortion order for B
B_0_2 = 2.31740244758215E-07 / Projection distortion parameter
B_0_3 = -1.39370219644737E-11 / Projection distortion parameter
B_0_4 = -1.93385788672272E-15 / Projection distortion parameter
B_1_1 = 1.24883116924871E-07 / Projection distortion parameter
B_1_2 = -1.75100337075546E-11 / Projection distortion parameter
B_1_3 = -6.68501352828139E-15 / Projection distortion parameter
B_2_0 = 2.32638864410737E-08 / Projection distortion parameter
B_2_1 = -4.31032879629697E-11 / Projection distortion parameter
B_2_2 = 1.47604729142386E-14 / Projection distortion parameter
B_3_0 = -1.57007398385189E-11 / Projection distortion parameter
B_3_1 = 1.99788395920582E-14 / Projection distortion parameter
B_4_0 = 2.58969983594055E-14 / Projection distortion parameter
B_DMAX = 1.29263602552885 / Projection distortion parameter
AP_ORDER= 4 / Distortion order for AP
AP_0_1 = 2.97065408706731E-08 / Projection distortion parameter
AP_0_2 = -4.75283677350868E-08 / Projection distortion parameter
AP_0_3 = -1.10315805523E-11 / Projection distortion parameter
AP_0_4 = 4.55114647901708E-15 / Projection distortion parameter
AP_1_0 = -3.58786283508571E-08 / Projection distortion parameter
AP_1_1 = -1.96188206395097E-07 / Projection distortion parameter
AP_1_2 = 2.37408989341839E-11 / Projection distortion parameter
AP_1_3 = 2.53456154277969E-15 / Projection distortion parameter
AP_2_0 = -1.12429422103018E-07 / Projection distortion parameter
AP_2_1 = 2.1567428794094E-11 / Projection distortion parameter
AP_2_2 = 4.11777831181248E-15 / Projection distortion parameter
AP_3_0 = 4.14362825790236E-11 / Projection distortion parameter
AP_3_1 = -1.65922157775061E-14 / Projection distortion parameter
AP_4_0 = -4.2864782379355E-14 / Projection distortion parameter
BP_ORDER= 4 / Distortion order for BP
BP_0_1 = -1.81685424974621E-08 / Projection distortion parameter
BP_0_2 = -2.3173422613022E-07 / Projection distortion parameter
BP_0_3 = 1.40445383648226E-11 / Projection distortion parameter
BP_0_4 = 1.91581457234528E-15 / Projection distortion parameter
BP_1_0 = -4.08016559219626E-08 / Projection distortion parameter
BP_1_1 = -1.24912850875228E-07 / Projection distortion parameter
BP_1_2 = 1.76186546854184E-11 / Projection distortion parameter
BP_1_3 = 6.65875991265602E-15 / Projection distortion parameter
BP_2_0 = -2.32881572697355E-08 / Projection distortion parameter
BP_2_1 = 4.3272203917565E-11 / Projection distortion parameter
BP_2_2 = -1.48095849601585E-14 / Projection distortion parameter
BP_3_0 = 1.58118070097522E-11 / Projection distortion parameter
BP_3_1 = -2.0006530867698E-14 / Projection distortion parameter
BP_4_0 = -2.59146367997654E-14 / Projection distortion parameter

ACKNOWLEDGMENTS

PTF is a scientific collaboration between the California Institute of Technology, Columbia University, Las Cumbres Observatory, the Lawrence Berkeley National Laboratory, the National Energy Research Scientific Computing Center, the University of Oxford, and the Weizmann Institute of Science.

This interproject work was performed at the Spitzer Science Center and the NASA Herschel Science Center as part of missions/projects managed by Jet Propulsion Laboratory, California Institute of Technology, under contracts with the National Aeronautics and Space Administration.

REFERENCES

1. N. M. Law, S. R. Kulkarni, R. G. Dekany, E. O. Ofek, R. M. Quimby, P. E. Nugent, J. Surace, C. J. Grillmair, J. S. Bloom, M. M. Kasliwal, L. Bildsten, T. Brown, B. Cenko, D. Ciardi, E. Croner, S. G. Djorgovski, J. van Eyken, A. V. Filippenko, D. B. Fox, A. Gal-Yam, D. Hale, N. Hamam, G. Helou, J. Henning, D. A. Howell, J. Jacobsen, R. Laher, S. Mattingly, D. McKenna, A. Pickles, D. Poznanski, G. Rahmer, A. Rau, W. Rosing, M. Shara, R. Smith, D. Starr, M. Sullivan, V. Velur, R. Walters, and J. Zolkower, “The Palomar Transient Factory: system overview, performance, and first results,” *PASP* **121**, pp. 1395–1408, 2009.
2. E. W. Greisen and M. R. Calabretta, “Representations of world coordinates in FITS,” *Astronomy & Astrophysics* **395**, pp. 1061–1075, 2002.
3. M. R. Calabretta and E. W. Greisen, “Representations of celestial coordinates in FITS,” *Astronomy & Astrophysics* **395**, pp. 1077–1122, 2002.
4. E. Bertin, “Automatic astrometric and photometric calibration with SCAMP,” *ASP Conf. Ser.* **351**, p. 112, 2006.
5. E. Bertin and S. Arnouts, “SExtractor: software for source extraction,” *Astron. Astrophys. Suppl. Ser.* **117**, pp. 393–404, 1996.
6. E. Bertin, Y. Mellier, M. Radovich, G. Missonnier, P. Didelon, and B. Morin, “The TERAPIX Pipeline,” *ASP Conf. Ser.* **281**, pp. 228–237, 2002.
7. D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis, “Astrometry.net: blind calibration of arbitrary astronomical images,” *AJ* **139**, pp. 1782–1800, 2010.
8. D. L. Shupe, M. Moshir, J. Li, D. Makovoz, and R. Narron, “The SIP convention for representing distortion in FITS image headers,” *ASP Conf. Ser.* **347**, pp. 491–495, 2005.
9. R. R. Laher, V. Gorjian, L. M. Rebull, F. J. Masci, J. W. Fowler, G. Helou, S. R. Kulkarni, and N. Law, “Aperture Photometry Tool,” *PASP manuscript in press*, 2012.
10. D. Makovoz and F. R. Marleau, “Point-source extraction with MOPEX,” *PASP* **117**, pp. 1113–1128, 2005.
11. A. S. Fruchter and R. N. Hook, “Drizzle: A method for the linear reconstruction of undersampled images,” *PASP* **114**, pp. 144–152, 2002.
12. F. J. Masci and J. W. Fowler, “AWAIC: a WISE astronomical image co-adder,” *ASP Conf. Ser.* **411**, pp. 67–76, 2009.