Astrometry in Wide-Field Surveys

 András Pál
 Department of Astronomy, Loránd Eötvös University, H-1117 Budapest, Hungary; apal@cfa.harvard.edu

 AND

 Gáspár Á. Bakos
 Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138

 Received 2006 July 21; accepted 2006 August 22; published 2006 October 26

 ABSTRACT. We present a robust and fast algorithm for performing astrometry and source cross-identification on lists of two-dimensional points, such as between a catalog 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 a large number of sources and significant nonlinear 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 extended Delaunay triangulation. Our software implementation performed at the 99.995% success rate on ∼260,000 frames taken by the HATNet project.

1. INTRODUCTION

Cross-matching two lists of two-dimensional points is a crucial step in astrometry and source identification. The task involves finding the appropriate geometric transformation that transforms one list into the reference frame of the other, and then finding the best matching point pairs. One of the lists usually contains the pixel coordinates of sources in an astronomical image (e.g., pointlike sources, such as stars), while the other list can either be 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 paper, we denote the reference (list) as $R$, the image (list) as $I$, and the function that transforms the reference to the image as $F_{R\rightarrow I}$.

The difficulty of the problem is that in order to find matching pairs, one needs to know the transformation, and vice 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 $F_{R\rightarrow I}$, however, the problem can be tackled. A very specific case is one in which 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 the method proposed by Thiebaut et al. (2001) uses all of the image information to derive a transformation (translation and magnification).

A more general assumption typical of astronomical applications is that $F_{R\rightarrow I}$ is a similarity transformation (rotation, magnification, and inversion, without shear); i.e., $F_{R\rightarrow I} = \lambda A r + b$, where the matrix $A$ is a (nonzero) scalar times the orthogonal matrix, $b$ is an arbitrary translation, and $r$ is the spatial vector of points. Exploiting the fact that geometric 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 with the aid of a specific set of patterns that are generated from a subset of the points on both $R$ and $I$. 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 IMMATCH task (Phillips & Davis 1995) in IRAF.3

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 below). In the

---

1 Visiting Astronomer, Harvard-Smithsonian Center for Astrophysics.
2 Hubble Fellow.

3 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.
past decade of astronomy, with the development of large-format CCD cameras or mosaic imagers, many wide-field surveys have appeared, such as those looking for transient events (e.g., ROTSE; Akerlof et al. 2000), transiting planets (e.g., KELT, Pepper et al. 2004; TrES, Alonso et al. 2004, HATNet, Bakos et al. 2002, 2004; see Charbonneau et al. 2006 for further references), or all-sky variability (e.g., ASAS; Pojmanski 1997). There are nonnegligible, higher order distortion terms in the astrometric solution that are due, for instance, to the projection of celestial to pixel coordinates and the properties of the fast-focal-ratio optical systems. Furthermore, these images may contain \( \sim 10^7 \) sources, and pattern matching is nontrivial.

These surveys have necessitated a further generalization of the algorithm, which we present in this paper. More specifically, we were motivated by the astrometric requirements of the Hungarian-made Automated Telescope Network (HATNet). Each HAT telescope in the network consists of a 200 mm focal length, f/1.8 telephoto lens and a 2K \( \times \) 2K CCD yielding an 8\(^{\circ} \times 8\(^{\circ} \) FOV. In our experience, we need at least fourth-order polynomial functions of the pixel coordinates in order to properly describe the distortion of the lens. With a typical exposure time of 5 minutes in the I band, in a moderately dense field (\( b \approx 15^{\circ} \)), there are 30,000 stars brighter than \( I = 13 \) for which better than 10\% photometry can be achieved. If we consider all 3 \( \sigma \) detections, we have to deal with the identification of \( \sim 100,000 \) sources.

The algorithm presented in this paper 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 (1) the distortions are nonnegligible but small compared to the linear term, (2) there exists a smooth transformation between the reference and image points, (3) the point lists have a considerable number of sources in common, and (4) the transformation is locally invertible. The paper is presented as follows. First, we describe symmetric point matching in \( \S \) 2, followed by a discussion of finding the transformation (\( \S \) 3). The software implementation and its performance on a large and inhomogeneous data set is demonstrated in \( \S \) 4. Finally, we draw conclusions in \( \S \) 5.

2. SYMMETRIC POINT MATCHING

First, let us assume that \( F_{R \rightarrow I} \) is known. To find point pairs between \( R \) and \( I \), one should first transform the reference points to the reference frame of the image: \( R' = F_{R \rightarrow I}(R) \). Now it is possible to perform a simple symmetric point matching between \( R' \) and \( I \). One point \( (R_i \in R') \) from the first and one point \( (I_i \in I) \) from the second set are treated as a pair if the closest point to \( R_i \) is \( I_i \) and the closest point to \( I_i \) is \( R_i \). This requirement is symmetric by definition and excludes such cases in which, e.g., the closest point to \( R_i \) is \( I_i \) but there exists an \( R_j \) that is even closer to \( I_i \), 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. There is to be done only once at the beginning, and using the QuickSort algorithm; for example, the required time scales, on average, as \( O(N \log N) \). 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. In 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 \( O(\log N) \) function of \( N \). Thus, the total time, including the initial sorting, also goes as \( O(N \log N) \).

As regards a list of two-dimensional points, let us assume again that the points are given in ascending order by their \( x \)-coordinates [initial sorting \( \sim O(N \log N) \)], and that they are spread uniformly in a square of unit area. Finding the nearest point to an \( x \)-coordinate also requires \( O(\log N) \) comparisons; however, the point that is found presumably will not be 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 that have this width and unity height, meaning \( O(\sqrt{N}) \) comparisons. Therefore, the total time required for a symmetric point matching between two catalogs in two dimensions requires \( O(N^{1/2} \log N) \) time.

We note that finding the closest point within a given set of points is also known as the nearest neighbor problem (for a summary, see Gionis 2002 [unpublished] and references therein). \(^4\) It is possible to reduce the computation time in two dimensions to \( O(N \log N) \) with the aid of Voronoi diagrams and cells, but we have not implemented such an algorithm in our matching codes.

3. FINDING THE TRANSFORMATION

Let us return to the task of finding the transformation between \( R \) and \( I \). The first and most crucial step of the algorithm is to find an initial “guess” \( F_{R \rightarrow I}^{(0)} \) for the transformation, based on a variant of triangle matching. Using \( F_{R \rightarrow I}^{(0)} \), \( R \) is transformed to \( I \), symmetric point matching is done, and the paired coordinates are used to further refine the transformation [leading to \( F_{R \rightarrow I}^{(i)} \) in iteration \( i \)] and increase the number of matched points iteratively. A major part of this paper is devoted to finding the initial transformation.

3.1. Triangle Matching

It was proposed earlier by Groth (1986) and Stetson (1989), and recently by others (see Valdes et al. 1995), that triangle matching be used 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 \( O(N^3) \) function of \( N \). As this can be an overwhelming number, one can resort to using a subset of

---

\(^4\) See http://theory.stanford.edu/~nmishra/CS361-2002/lecture12-scribe.pdf.
the points for the vertices of the triangles that are to be generated. One can also limit the parameters of the triangles, such as excluding 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 three-dimensional triangle space. Because our assumption is that it is dominated by the linear term, to first-order approximation there is a single scalar magnification between \( R \) and \( I \) (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 \( R \) and \( I \).

### 3.1.1. Triangle Spaces

There are multiple ways of deriving normalized triangle spaces. One can define a “mixed” normalized triangle space \( T^{\text{mix}} \) in which 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:

\[
T_{x}^{\text{mix}} = p/la, \quad (1)
\]
\[
T_{y}^{\text{mix}} = q/la, \quad (2)
\]

(Valdes et al. 1995), where \( a, p, \) and \( q \) are the sides of the triangle, in descending order. Triangles in this space are shown in the left panel of Figure 1. 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 \( F_{R,I} \) is smooth and bijective, no local inversions and flips can occur. In other words, \( R \) and \( I \) 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, repeat the fit with flipped triangles.

This leads to the definition of an alternative “chiral” triangle space:

\[
T_{x}^{\text{chir}} = b/la, \quad (3)
\]
\[
T_{y}^{\text{chir}} = c/la, \quad (4)
\]

where \( a, b, \) and \( c \) are the sides in counterclockwise 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 parameterization that is both continuous and that preserves chirality. Flip the chiral triangle space in the right panel of Figure 1 along the \( T_x + T_y = 1 \) line. This transformation moves the equilateral triangle into the origin. Next, 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 \( x \)- and \( y \)-axis direction, and \( \sqrt{2} \) along the \( T_x = T_y \) line). Finally, apply an azimuthal slew by a factor of \( \pi/4 \) to identify the \( T_x = 0, T_y > 0 \) and \( T_y = 0, T_x > 0 \) edges of the space. To be more specific, let us denote the sides as in \( T^{\text{chir}} \) as \( a, b, \) and \( c \), in counterclockwise order, where \( a \) is the longest, and define

\[
\alpha = 1 - b/a, \quad (5)
\]
\[
\beta = 1 - c/a. \quad (6)
\]

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

\[
x_1 = \frac{\alpha(\alpha + \beta)}{\sqrt{\alpha^2 + \beta^2}}, \quad (7)
\]
\[
y_1 = \frac{\beta(\alpha + \beta)}{\sqrt{\alpha^2 + \beta^2}}, \quad (8)
\]
\[
x_2 = x_1^2 - y_1^2, \quad (9)
\]
\[
y_2 = 2x_1y_1, \quad (10)
\]

one can define the triangle space coordinates as

\[
T_{x}^{\text{cont}} = \frac{x_2^2 - y_2^2}{(\alpha + \beta)^3} = \frac{(\alpha + \beta)(\alpha^4 - 6\alpha^2\beta^2 + \beta^4)}{(\alpha^2 + \beta^2)^3}, \quad (11)
\]
\[
T_{y}^{\text{cont}} = \frac{2x_2y_2}{(\alpha + \beta)^3} = \frac{4(\alpha + \beta)\alpha\beta(\alpha^2 - \beta^2)}{(\alpha^2 + \beta^2)^3}. \quad (12)
\]

The continuous triangle space \( T^{\text{cont}} \) defined here has many advantages. It is a continuous function of the sides for all nonsingular triangles, and it also preserves chirality information. Furthermore, it spans a larger area, and misidentification of triangles (which may be very densely packed) is decreased. Some triangles in this space are shown in Figure 2.

### 3.1.2. Optimal Triangle Sets

As mentioned above, 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 impractical, for the following reasons. First, storing and handling such a large number of triangles with typical computers is inconven-
Fig. 1.—Position of triangles in mixed and 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 have identical side ratios but different orientation overlap. The dashed line shows the boundaries of the triangle space. The dot-dashed line represents the right triangles and separates obtuse and acute triangles.

Fig. 2.—Triangles in the continuous triangle space as defined by eqs. (11)–(12). We show the same triangles as those in Fig. 1 for the \( T^{\text{mix}} \) and \( T^{\text{chir}} \) triangle spaces. Equilateral triangles are centered on the origin. The dot-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 \( \gamma^T \).

To make an estimate of the optimal size for triangles, let us use \( D \) to denote the characteristic size of the image, \( \delta \) for the astrometric error, and \( L \) as the size of a selected triangle. 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:

\[
f_d \approx \left| \sin d - \tan d \right|/d \approx \left| 1 - \cos d \right|
\]

(13)

(see Calabretta & Greisen 2002), where \( d \) is the radial distance as measured from the center of the field. For the HATNet frames \( d = D \approx 6^\circ \) 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 \( \delta L \) in the triangle space, which decreases with increasing \( L \). Making the two errors equal,

\[
\frac{f_d L}{D} = \frac{\delta}{L},
\]

an optimal triangle size can be estimated:

\[
L_{\text{opt}} = \sqrt{\frac{\delta D}{f_d}}.
\]

In our case, \( d = 2048 \) pixels (or \( 6^\circ \)), \( f_d = 0.005 \), 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 oversaturated by the large number of points and may yield unexpected matchings of triangles. In all definitions in the previous subsection, the area of the triangle space is approximately unity. Given triangles with an error \( \sigma \) in triangle space, assuming they have a uniform distribution with a \( 3 \sigma \) spacing between them, and assuming \( \sigma = \delta/L_{\text{opt}} \), the number of triangles is delimited to

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

(16)

In our case (see values of \( D \), \( f_d \), and \( \delta \) 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 [\( O(10^{11}) \)].

3.1.3. The Extended Delaunay Triangulation

Delaunay triangulation (see Shewchuk 1996) is a fast and robust way of generating a triangle mesh on a point set. Delaunay triangles are disjointed triangles in which the circumcircle of any triangle contains no 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 Figure 3 (left).

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

$$T_0 = 2N - 2 - C,$$

(17)

where $C$ is the number of edges on the convex hull of the point set. For large values of $N$, $T_0$ can be estimated as $2N$, as $2 + C$ is negligible. Therefore, if we select a subset of points (from $R$ or $I$) whose neighboring points are at a distance of $L_{\text{opt}}$, we get a Delaunay triangulation with approximately $2D^2L_{\text{opt}}^2$ triangles. The $D$, $\delta$, and $f_{\text{e}}$ values for HAT images correspond to $\approx 6000$ 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 to the removal of a point from a star list. According to the polyhedron formula, on average, each point has six neighboring points and belongs to six 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 $I$ 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 according to the changing FWHM of the frames. Transients, variable stars, or minor planets can lead to additional sources, on occasion. In general, if one point is removed, six Delaunay triangles are destroyed and four new ones form that are totally disjointed from the six original ones (and therefore are represented by substantially different points in the triangle space). Removing $\frac{1}{6}$ of the generating points might completely change the triangulation.$^5$

Second, and more important, is that there is no guarantee that the spatial density of points in $R$ and $I$ is similar. For example, the reference catalog is retrieved for stars with magnitude limits that are different from those found on the image. If the number of points in common in $R$ and $I$ is only a small fraction of the total number of points, the triangulations on the reference and image have an inappropriate number of (or even no) common triangles.

Third, the number of triangles with Delaunay triangulation $T_0$ is definitely smaller than $T_{\text{opt}}$, i.e., the triangle space could support more triangles without much confusion.

Therefore, it is beneficial to extend the Delaunay triangulation. A natural way to do this is as follows. Define a level $\ell$, and for any given point $P$ select all points from the set of $N$ points that can be connected to $P$ via maximum $\ell$ 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 $\ell$. For self-consistence, the $\ell = 0$ case is defined as the Delaunay triangulation itself. If all points have six neighbors, the number of “extended” triangles per data point is

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

(18)

for $\ell > 0$; i.e., this extension introduces $O(\ell^6)$ new triangles. Because some of the extended triangles are repetitions of other triangles from the original Delaunay triangulation and from the extensions of other points, the final dependence only goes as $O(T_{\ell}\ell^2)$. We note that our software implementation is slightly different, and the expansion requires $O(N\ell^2)$ time and automatically results in a triangle set in which 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, $\ell = 2$ gives some $\sim 347,000$ triangles, $\ell = 3$ gives 875,000 triangles, and $\ell = 4$ gives $\sim 1,841,000$ triangles. The extended triangulation is advantageous not only because it provides more triangles and thus has a better chance for matching, but also because there is a bigger variety in size that enhances matching if the input and reference lists have different spatial densities.

3.1.4. Matching the Triangles in Triangle Space

If the triangle sets for both the reference and 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 § 2.

In the next step, we create an $N_r \times N_i$ “vote” matrix $V$, where $N_r$ and $N_i$ are the number of points in the reference and input lists, respectively, that were used to generate the triangulations. The elements of this matrix have an initial value of zero. Each matched triangle corresponds to three points in the reference list (identified by $r_1$, $r_2$, and $r_3$) and three points in the input list ($i_1$, $i_2$, and $i_3$). Knowing these indices, the matrix elements $V_{r_1i_1}$, $V_{r_2i_2}$, and $V_{r_3i_3}$ are incremented. The magnitude

---

$^5$ Imagine a honeycomb structure in which all central points of the hexagons are added or removed; these two constructions generate disjoint Delaunay triangulations.
of this increment (the “vote”) can depend on the distances of the matching triangles in the triangle space: the closer they are, the more votes these points get. In our implementation, if \( N_T \) triangles are completely matched, 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 numbers of votes are likely to be due to misidentifications. We found that in practice, the upper 40% of the matrix elements yield a robust match.

### 3.2. The Unitarity of the Transformations

If an initial set of 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. Prior to the transformation, our assumption is that the dominant term in the transformation is the similarity transformation, which implies that the homogeneous linear part of it should almost be a unitarity operator. After the transformation is determined, it is useful to measure how much we diverge from this assumption. As mentioned above (§ 1), similarity transformations can be written as

\[
r' = \lambda Ar + b = \lambda \begin{pmatrix} a & c \\ b & d \end{pmatrix} r + b,
\]

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

If we separate the homogeneous linear part of the transformation, as described by a matrix similar to that in equation (19), it will be a combination of rotation and dilation with possible inversion if \( |a| \approx |d| \) and \( |c| \approx |b| \). We can define the unitarity of a \( 2 \times 2 \) matrix as

\[
\Lambda^2 = \frac{(a \mp d)^2 + (b \pm c)^2}{a^2 + b^2 + c^2 + d^2},
\]

where the plus (minus) indicates the definition for regular (inverting) transformations, respectively. For a combination of rotation and dilation, \( \Lambda \) is zero, and for a distorted transformation, \( \Lambda \approx f_0 \ll 1 \).

The \( \Lambda \) 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 \( \Lambda \) is not just larger than the expectational value of \( f_0 \), but is \( \approx 1 \). This enables fine-tuning of the algorithm, such as changing chirality of the triangle space or adding further iterations until satisfactory \( \Lambda \) is reached.

### 3.3. Point Matching in Practice

In practice, matching points between the \( R \) reference and \( I \) image proceeds as follows:

1. Generate two triangle sets \( T_R \) and \( T_I \) on \( R \) and \( I \), respectively:
   - \( a \) In the first iteration, generate only Delaunay triangles.
   - \( b \) Later, if necessary, extended triangulation can be generated with increasing levels of \( \ell \).
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 \( N_p \) pairs).
4. Derive the initial smooth transformation \( F_{R 
 leftarrow I}^{(0)} \), using a least-squares fit.
   - \( a \) Check the unitarity of \( F_{R 
 leftarrow I}^{(0)} \).
   - \( b \) If it is greater than a given threshold \( O(f_0) \), increase \( \ell \) and go to step \( 1b \). If the unitarity is less than this threshold, proceed to step \( 5 \).
   - \( c \) If the maximal allowed \( \ell \) is reached, try the procedure with triangles that are flipped with respect to each other between the image and reference (i.e., switch chirality of the \( T^{(cont)} \) triangle space).
5. Transform \( R \) using this initial transformation to the reference frame of the image \( [R 
 leftarrow I] \).
6. Perform a symmetric point matching between \( R' \) and \( I \) (yielding \( N_j > N_p \) pairs).
7. Refine the transformation based on the greater number of pairs, yielding transformation \( F_{R 
 leftarrow I}^{(i)} \), 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 \( R \) and \( I \), 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 because local triangles are used, it proves to be a robust procedure. It should be emphasized that \( F_{R 
 leftarrow I}^{(i)} \) can be any smooth transformation; for example, an affine transformation with small shear, or a polynomial transformation of any reasonable order. The optimal value of the order depends on the magnitude of the distortion. Detailed descriptions of fitting procedures for such models and functions can be found in various textbooks (see, e.g., Press et al. 1992, chapter 15). It is noteworthy that in step 7, one can perform a weighted fit with possible iterative rejection of \( n \sigma \) outlier points.
4. SOFTWARE IMPLEMENTATION AND APPLICATIONS

4.1. Software Implementation

The coordinate matching and transformation algorithms are implemented in two stand-alone binary programs written in ANSI C. The program grmatch matches point sets, including triangle space generation, triangle matching, symmetric point matching, and polynomial fitting; that is, steps 1–4 in § 3.3. The other program, grtrans, transforms coordinate lists using the transformation coefficients that are outputted 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 external software. We should note that in the case of degeneracy (e.g., when all points are on a perfect lattice), the match will fail.

Both programs are part of the FIHAT/HATpipe package that is under development for the massive data reduction of the HATNet data flow. They can be easily embedded into UNIX environments, as both of them parse a wide range of command-line arguments for defining the structure of the input data and fine-tuning the algorithm. The programs are also capable of redirecting their input and output to standard streams.

By combining grmatch and grtrans, one can easily derive the World Coordinate System (WCS) information for a FITS data file. The output of WCS keywords is now fully implemented in grtrans, following the conventions of the WCSTools package (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 the WCS, see Calabretta & Greisen (2002), and for the representation of distortions, see Shupe et al. (2005).

The package containing the programs grmatch and grtrans and other related software are accessible online (after registration).

4.2. Performance on Large Data Sets

We used grmatch and grtrans to perform astrometry and star identification on a large set of images taken by the HAT Network of telescopes (Bakos et al. 2004). The results presented in this paper are based on observations originating from the following HATNet telescopes: HAT-5, HAT-6, and HAT-7, located at the Fred Lawrence Whipple Observatory (FLWO), Arizona, plus HAT-8 and HAT-9 on the Smithsonian Submillimeter Array roof, atop Mauna Kea, Hawaii. Briefly, the survey telescopes have an identical setup: a 200 mm f/1.8 telephoto lens and a 2K CCD yielding an 8° × 8° FOV.

In order to test the method on different instruments, we also performed astrometry on data taken by the follow-up instrument TopHAT (FLWO). TopHAT is a 0.26 m diameter, f/5 Ritchey-Chrétien design telescope with a Baker wide-field corrector aided by a 2K × 2K Marconi chip, yielding a 173 FOV.

The astrometry and identification steps were as follows. First, for all observed fields, reference star lists were generated using the Two Micron All Sky Survey catalog (2MASS; see Skrutskie et al. 2006) as reference. These reference lists include the source identifiers, the original celestial coordinates (R.A., decl.), an estimated J-band magnitude, and the projected coordinates (ξ, η) of the stars. We used arc projection (see Calabretta & Greisen 2002) centered at the nominal center of a given field, and scaling of the projection was unity such that a star located at a distance of 1° from the center of the given celestial field had a unit distance in the (ξ, η) plane from the origin in the reference list. The FOVs of the reference lists were a bit wider than the nominal FOVs of the HAT telescopes, to ensure a complete overlap between the two lists in spite of the small uncertainties in the positioning of the telescopes.

Second, an input star list was generated for each image, using our star detection algorithm fstar (also part of FIHAT/HATpipe), which detects and fits starlike objects above a given signal-to-noise ratio threshold. This detection yields a set of input lists that include the pixel coordinates (X, Y) of the stars, and other quantities (including the flux, FWHM, and the shape parameters).

Third, for each image, the input star list and the relevant reference star list were matched using the program grmatch. The match was performed between the projected reference coordinates (ξ, η) and the detected pixel coordinates (X, Y). The program outputs two files: the list of the matched lines (the “match” file) and a small file that includes the fitted polynomial transformation parameters and some statistical data (the “transformation” file). It should be emphasized that the match was not done directly using the original celestial coordinates, as they exhibit an unwanted curvature in the field.

Finally, the reference star list (ξ, η) was transformed by the program grtrans into the system of the (X, Y) image, using the “transformation” file. The transformed list shows where each star with a given identifier would fall on the image. The transformation can also be used to calculate the WCS information for a given image.

We note that the crucial part of the process is the third step. This can be fine-tuned by using many parameters, one of the most important being the polynomial order. For a small FOV (less than 1°) and small distortions, linear or second-order polynomials yield good results. For HAT images, we had to increase the order up to 6 to achieve the best results. Figure 4 exhibits two vector plots that show the difference between the transformed reference coordinates and the detected star coordinates using a second- and a fourth-order polynomial transformation for a typical HAT image. In the first case, by using a second-order fit, definite radial structures remain, and the stars located at the corners of the image are not evenly matched, due to the large distortions in the optics. However, using a
fourth-order fit, all segments of the image are matched, and the residuals are also smaller. These small residuals can be better visualized if only the difference between one of the coordinates is shown in a gradient plot. Figure 5 illustrates the difference between the $Y$-coordinates for the same image, using a fourth- and sixth-order polynomial fit. While there is a definite residual structure in the fourth-order fit, it disappears using the sixth-order polynomial transformation.

As regards statistics, we performed the astrometry and source identification for 243,447 HAT images that had been acquired between the beginning of 2003 and 2006 June. The wide-field telescopes observed 52 individual and almost non-overlapping fields between the Galactic latitudes $b = -30^\circ$ and $b = +74^\circ$.

We initiated the processing with the following parameters. For the triangulation, the 3000 brightest sources were used from both the reference catalog and the detected stars. The critical unitarity was set to 0.01; therefore, if the fitted initial transformation had a unitarity larger than this value, the level of the triangulation expansion was increased. The final transformation was determined using a weighted sixth-order polynomial fit. Because the astrometric errors of brighter stars are smaller, we weighted data points based on their magnitude during the fit. Finally, the maximal distance of matches was set to 1 pixel to reject false identifications.

Astrometry and cross-identification of sources was successful for 238,353 images. The remaining 5094 images were analyzed manually, and we found that only 13 of them were good enough to expect astrometry to succeed; the rest were cloudy or showed various other errors. Astrometry on these 13 images also succeeded by decreasing the number of stars for triangulation to 2000. This means that a completely automatic run yielded a 99.995% success rate, and the other images were also matched by applying small changes to the fine-tuned parameters.

In order to test the algorithm with a different instrument, we also performed astrometry on 22,936 TopHAT images taken in 2005. The only difference in the procedure was that the polynomial transformation was only of second order. The success ratio was 93%, but 90% of the frames in which astrometry failed were cloudy, with virtually no stars. Astrometry also failed on very short exposure (10 s) $V$-band frames. Fine-tuning the parameters (number of triangles, input lists) resolved most of these cases.

The following statistics were compiled on the wide-field HAT frames. The median number of matched sources relative to the number of stars in the reference or the input list was 98.38% ± 0.31% (median deviation). The average CPU usage was $0.77 \pm 0.22 \text{ s frame}^{-1}$ on a 64 bit AMD Opteron machine running at 2 GHz. Astrometry was successful on 96.78% of the images using Delaunay triangulation without extended triangles (CPU time: 0.73 s); 0.49% of the frames were processed at level $\ell = 1$ extended triangulation (CPU: 1.79 s), 0.06% at $\ell = 2$ (CPU: 2.22 s), 33 images at $\ell = 3$ (CPU: 3.67 s), and 1334 + 5094 images at $\ell = 4$ (CPU: 5.20 s). Here the number 5094 refers to those images for which astrometry failed even at $\ell = 4$, mostly because of bad data quality (see above). The reason for the success of Delaunay triangulation for 96% of the wide-field HAT frames without extended triangulation is because the HAT instruments perform homogeneous data acquisition and are very well characterized (zero points, saturation). Thus, the 2MASS reference catalogs can be retrieved for a given field in such a way that there are many sources in common. However, in general applications, when the saturation and faint magnitude limits of an image have only a crude estimate, extended triangulation is essential.

Although the procedure is fast, we note that the most time-consuming part of the process is the triangulation generation and the triangle matching itself. On average, this required more than 60% of the total time, and at $\ell = 4$, 92% of the time. The median value of the fit residuals was 0.06 pixels, while the median of the unitarities was 0.0042. The latter is in quite good agreement with the expected value of the nonlinearity factor, $f_n \approx 0.005$.

### 4.3. Comparison with Other Implementations

We also compared the performance of the program `gmatch` with an existing implementation within IRAF,
namely the IMAGES.IMMATCH package, using the related tasks \textit{xyxymatch}, \textit{geomap}, and \textit{geoxytran}. The steps of the point matching were as described in § 3.3. First, an initial set of possible pairs were established using \textit{xyxymatch} and the “triangles” option as the matching method. Because the triangle sets generated by \textit{xyxymatch} are full triangulations, we limited our input lists to the brightest sources; otherwise, the $O(N^2)$ dependence of the number of triangles would have resulted in an unrealistically long matching time. Second, the initial transformation was fitted using \textit{geomap}, followed by a transformation of the reference catalog to the frame of the input list using \textit{geoxytran} and this fit. Third, the transformed reference and the original input list were also matched by \textit{xyxymatch}, but this time using the “tolerance” matching method. Finally, this new list of point pairs was used again to refine the geometric transformation with \textit{geomap}.

The comparison between grmatch and the IRAF IMAGES.IMMATCH implementation was based on 950 individual images, all acquired by TopHAT from the same FOV. We note that we had to use the relatively narrow-field TopHAT for the comparison, as the triangle match on the original 8’2 HATNet frames is almost hopeless, given the spatial distortions, the large number of stars, and the difficulty of selecting the brightest stars and at the same time retaining a small total number of selected sources (in order to be able to cope with a full triangulation). On each image, there were approximately 800–900 detected stars, depending on the air mass or thin clouds. For the triangulation and the initial \textit{xyxymatch} fit, we used the 35 brightest sources from both the reference catalog and the input star lists. We found that \textit{grmatch} required $\sim$0.1 s CPU time on average, while the whole procedure using the IRAF-based tasks, as described above, required $\sim$5–7 s net CPU time for a single image. Both algorithms yielded the same transformation coefficients and found the same number of pairs. However, in three cases, the number of sources used for triangulation had to be increased manually to 40 or 45. It is noteworthy that although the IRAF version proved to be significantly slower, the time-consuming part was the first \textit{xyxymatch} matching. All other tasks, including the second matching (with “tolerance” option), required only a fraction of a second per image.

5. SUMMARY

In this paper, we present a robust algorithm for cross-matching two lists of two-dimensional points. The task is twofold: finding the smooth spatial transformation between the lists, and cross-matching the points. These two steps are intertwined and are performed in an iterative way until a satisfactory transformation and matching rate are reached. We make only very basic assumptions that hold for almost all astronomical applications, including wide-field surveys with distorted fields and a large number of sources. Namely, the transformation between the point lists is largely a similarity transformation (arbitrary shift, rotation, magnification, inversion). A significant distortion term can be present, given that it can be linearized on the scale length of the average distance of neighboring points.

In § 2 we briefly describe symmetric point matching in one and two dimensions, because this tool is used throughout the astrometry procedure. Finding the initial transformation between the point lists is based on triangle matching. First, we define various normalized triangle spaces in § 3.1.1. The “mixed” triangle space of Valdes et al. (1995) is a continuous function of the triangle parameters, but flipped triangles are not distinguished. The “chiral” triangle space ensures that chirality information is preserved, but this space is not continuous. We show that it is possible to define a “continuous” triangle space that is both continuous and preserves chirality, and that furthermore spans a larger volume and diminishes the confusion of triangles with similar coordinates.

Taking into account the distortion of a field and the astrometric errors, we calculate both the optimal size and number of triangles. For the typical setup of a HATNet telescope (8° × 8° FOV, distortion factor $f_0 \sim 0.005$), the optimal size is 02, and the optimal number of triangles is less than 2 × 103. We use Delaunay triangulation for generating the triangles of the triangle space. This has the advantage of being fast, robust, and generating local triangles that are less prone to being distorted. However, we note in § 3.1.3 that Delaunay triangulation is sensitive to the removal or addition of points to the list, and thus is unstable. We introduce an extension of this triangulation that is parameterized by an $\ell$ level.

Having determined the transformation between the two lists, it is possible to check how well the initial assumption about the linearity of the dominant term holds. In § 3.2, we introduce the unitarity of the transformation, a simple scalar measure of this property. We describe the practical details of the algorithm in § 3.3 and the actual software implementation (\textit{grmatch}, \textit{grtrans}) in § 4.1.

Finally, we ran these programs on some 240,000 frames taken by the wide-angle cameras of HATNet, plus 20,000 frames acquired by the TopHAT telescope. The success rate was very close to 100%, and the routines handled the various pointing errors, defocusing, and sixth-order distortions in the wide fields. Both programs will become available from the authors upon request, in binary format and for a wide range of architectures.

A. P. would like to thank the hospitality of the Harvard-Smithsonian Center for Astrophysics, where this work was partially carried out. A. P. was also supported by Hungarian OTKA grant T-038437. The HATNet project is funded by NASA grant NNG 04-GN74G. G. Á. B. wishes to acknowledge funding from NASA Hubble Fellowship grant HST-HF-01170.01-A. Both authors would like to thank István Domsa for the early development of triangle and point-matching codes.
REFERENCES

Akerlof, C., et al. 2000, AJ, 119, 1901
Alonso, R., et al. 2004, ApJ, 613, L153
Bakos, G. A., Lázár, J., Papp, I., Sári, P., & Green, E. M. 2002, PASP, 114, 974
Bakos, G. Á., Noyes, R. W., Kovács, G., Stanek, K. Z., Sasselov, D. D., & Domsa, I. 2004, PASP, 116, 266
Calabretta, M. R., & Greisen, E. W. 2002, A&A, 395, 1077
Charbonneau, D., Brown, T. M., Burrows, A., & Laughlin, G. 2006, in Protostars and Planets V, ed. B. Reipurth, D. Jewitt, & K. Keil (Tucson: Univ. Arizona Press), in press (astro-ph/0603376)
Groth, E. J. 1986, AJ, 91, 1244
Joye, W. A., & Mandel, E. 2003, in ASP Conf. Ser. 295, Astronomical Data Analysis Software and Systems XII, ed. H. E. Payne, R. I. Jedrzejewski, & R. N. Hook (San Francisco: ASP), 489
Mink, D. J. 2002, in ASP Conf. Ser. 281, Astronomical Data Analysis Software and Systems XI, ed. D. A. Bohlender, D. Durand, & T. H. Handley (San Francisco: ASP), 169
Pepper, J., Gould, A., & Depoy, D. L. 2004, in AIP Conf. Proc. 713, The Search for Other Worlds (New York: AIP), 185
Phillips, A. C., & Davis, L. E. 1995, in ASP Conf. Ser. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco: ASP), 297
Pojmanski, G. 1997, Acta Astron., 47, 467
Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in C: The Art of Scientific Computing (2nd ed.; Cambridge: Cambridge Univ. Press)
Shewchuk, R. J. 1996, in Applied Computational Geometry: Towards Geometric Engineering, ed. M. C. Lin & D. Manocha (Berlin: Springer), 1148, 203
Shupe, D. L., Moshir, M., Li, J., Makovoz, D., Narron, R., & Hook, R. N. 2005, in ASP Conf. Ser. 347, Astronomical Data Analysis Software and Systems XIV, ed. P. Shopbell, M. Britton, & R. Ebert (San Francisco: ASP), 491
Skrutskie, M. F., et al. 2006, AJ, 131, 1163
Stetson, P. B. 1989, in V Advanced School of Astrophysics, ed. B. Barburi et al. (São Paulo: Univ. São Paulo)
Valdes, F. G., Campusano, L. E., Velásquez, J. D., & Stetson, P. B. 1995, PASP, 107, 1119
Thiebaut, C., & Boë, M. 2001, in ASP Conf. Ser. 238, Astronomical Data Analysis Software and Systems X, ed. F. R. Harnden, Jr., F. A. Primini, & H. E. Payne (San Francisco: ASP), 388