In press: Publications of the Astronomical Society of the Pacific.

A New Approach to Point Pattern Matching

F. MURTAGH
Space Telescope – European Coordinating Facility,
European Southern Observatory,
Karl-Schwarzschild-Str. 2,
D-8046 Garching (Germany).

Abstract

We describe a new algorithm for matching star lists, given by their 2-dimensional coordinates. Such matching should be unaffected by translation, rotation, rescaling, random perturbations, and some random additions and deletions of coordinate couples in one list relative to another. The first phase of the algorithm is based on a characterization of a set of coordinate couples, relative to each individual coordinate couple. In the second phase of the algorithm, the matching of stars in different lists is based on proximity of feature vectors associated with coordinate couples in the two lists. The order of magnitude computational complexity of the overall algorithm is n^2, for $O(n)$ coordinate couples in the coordinate lists.

Keywords: data-handling techniques – two-dimensional coordinate lists – matching stars – cross-identification

1 Introduction

The problem of point pattern matching arises in 2-dimensional photometry, and in matching star lists against catalog information. The former area will be most at issue in this article. Star lists (i.e. centroids of star positions) can arise through different fitting procedures, or different 2-d photometry packages used on the same field; or through the reduction of images based on different color filters; or through reduction of partially-overlapping images taken at different times or with different detectors.

The term "point" (coordinates, ordinarily in 2-dimensional space) will be used for the star centroid, or central location, in this article. A matching of some points from one list with some points in the other list(s) is sought. Equivalently the transformation which optimally maps one list into the other is sought.

The next section reviews some approaches adopted for this problem, in astronomy and in computer vision generally.

2 A Short Review of Approaches

The matching of 2- or 3-dimensional points from two lists is a very common problem. Many further references can be found in the works cited in this section. Following a short review of diverse approaches which have been applied to this problem, we indicate some of the (minor) differences between this problem as it manifests itself in astronomical image processing, compared to other areas of machine vision. Finally, in this section, we describe approaches which have been used for the astronomical problem.

Umeyama (1991) discusses a least squares solution to transformations comprising rotation, translation and scaling, on a given set of points. Hence the point-set A, and the point-set B, have the same cardinality (i.e. number of points). The points can be m-dimensional, where integer m is possibly greater than 3. The optimization problem is set up, and solved, in matrix algebra terms. A number of earlier references are cited which solve this least squares problem for 3-dimensional data.

Griffin and Alexopoulos (1989) also seek a matching which is invariant to translation, rotation, scaling and noise, for point-sets A and B of the same cardinality. Firstly, the smallest enclosing circles for the two point-sets are obtained. Next, the centroids of both circles are determined. The translation between the two point-sets is determined from knowledge of these two centroids. In both A and B, points are then sorted lexicographically by polar angle (from a given horizontal axis) and distance from the centroid of the point-set. Conditions are given for the matching of points, using this information. In the case of noisy point positions, the problem is formulated as a maximum cardinality graph matching problem. In our approach, described later, we also use an ordered list of points, but these are ordered relative to each point in turn, rather than just from the overall centroid of the point-set.

In astronomy, as will be mentioned below, point-sets A and B are unlikely to be of identical cardinality. Ogawa (1986) considers lists A and B of differing cardinalities, i.e. if list B is derived from A, then some additions and deletions of points are allowed. His approach is invariant to translation, scale, and random perturbation, in addition to addition/deletion, given 2-dimensional point-sets. A Delaunay triangulation is used to tesselate the planes. Although computationally much less demanding than the triangulation-based approaches of Groth and Stetson (discussed below), we would question the sensitivity of a Delaunay triangulation, alone, for capturing the information inherent in real data sets which we have looked at. Ogawa's (1986) approach proceeds by matching triangles using "labels" (weights, e.g. astronomical magnitude ranges), leading to a consistency graph between point pairs. A maximal clique (maximal complete subgraph) is sought in this graph. The approach is illustrated on stellar constellations, including using a cylindrical projection of a given point-set.

Wong and Salay (1986) use the term "constellation" for point patterns in 3-d and stereoscopic vision. A branch-and-bound algorithm is used to expeditiously search all possible combinations of points, making use of a cost function based on pairwise Euclidean distances between points in one set and points in the other.

Parvin and Medioni (1989) set up the point pattern problem (for 3-d data in industrial vision) as a constraint-satisfaction problem. An objective function is formulated from the many constraints, and is solved using a Hopfield-Tank neural network approach. Parenthetically, it is possible, although not so far demonstrated for this problem, that the very efficient constraint-satisfaction approach currently used for Hubble Space Telescope observation (exposure) scheduling in the PEPSI system (see Johnston and Adorf 1991) would perform very well on such a formulation of the point pattern matching problem. A further reference to point pattern matching, using a Hopfield-Tank neural network approach, is Nasrabadi and Choo (1992). This latter reference obtains so-called "interesting points" from 2-dimensional digital images. These are salient points in the two images which can be more easily matched than other points. A matching of the 2-dimensional images is sought, based on the sets of "interesting points" derived from these images.

Chen and Huang (1991), in the context of determining 3-dimensional point correspondences in the study of motion, assume a rigidity constraint involving distances between points and angles between lines joining the points. They seek an unambiguous match, subject to such rigidity, and develop a least squares solution.

As mentioned in the above, the astronomical matching problem is characterized by (i) additions and deletions between the lists A and B, due to results of different color filters, etc.; (ii) potentially large numbers of points are at issue (although labeling, in the guise of stellar magnitudes, can allow selection); (iii) in common with other fields, invariance relative to translation, rotation, scaling, and small random errors, is sought. Magnitudes may be used not just for selection but furthermore astronomical matching may explicitly aim at a weighted matching, where the weights associated with points are magnitude-related. In 2-d photometry work, there may well be less need to consider a point-set of small cardinality (a "model", in the terminology of Ogawa 1984) to be matched against a point-set of larger cardinality (a "world"), as might be the case in industrial vision. This situation may be different when matching a set of points against catalog information. The very large set-cardinalities in question, here, require other solutions (cf. below in the treatment of Figures 5 and 6).

We now briefly review three approaches used in the astronomical context.

Routine `PAIR`, authored by A. Lauberts, has been in use in the European Southern Observatory for many years. It assumes a translation between A and B, only. The Euclidean distance between each point i in A and each point i' in B is determined. If $i' = i + c$, then $d^2(i, i') = c^2$ for matched points, and one would expect a spread distribution of values for distances with points i'' in B which ought not to be matched. Thus the mode of all pairwise distances (between all i in A and all i' in B) allows the value c to be determined. This approach is invariant to translation and random perturbation.

The use of a Delaunay triangulation to capture affine-invariant information on point pattern interrelationships has been mentioned. Groth (1986) implements an all-triangles matching approach. A range of speed-ups are applied to cut down on the matching of all triangles from the first list, i.e. $O(n^3)$, with a similar list from the second list. The principle efficiency tactic is to only match triangles with a ratio of longest side to shortest side which is within some tolerance. Groth finds the order of magnitude increase in computation to be improved from $O(n^6)$ to $O(n^{4.5})$. It is recommended that the number of points in both lists be limited to between 20 and 30 points for computational reasons, and such a selection may be carried out on the basis of the magnitudes associated with points.

Stetson (1990), discussing an algorithm he developed many years earlier, also implemented a triangle-based matching algorithm. Points are considered in order of decreasingly important magnitude. Following an initial matching of a small number (three) of highly-weighted points, further points are added one at a time. Hence the procedure, reasonably, is biased towards points of large weight (i.e. of important magnitude). Imprecision in measurement of magnitudes is taken into account to the extent that rank orders of magnitudes are used.

3 The Proposed Method

For each point, i, in either list, a "world view" vector is determined. This vector represents the $n-1$ other points in the same list, as "seen" from the point i. Relative to an initially arbitrary horizontal axis, the angles subtended by the $n-1$ other points to the given point, i, are determined, and sorted. We consider here, and in our experimentation, only 2-dimensional point-sets: angles could not be trivially sorted in dimensions higher than 3. At the angle subtended by j relative to i, we consider the effect of j as being related to $d(j,i)$, the usual Euclidean distance. We define the effect of j on i as $K - d(i,j)$ where K is a constant which is somewhat greater than the maximum $d(i,j)$ for all i,j. For scale independence, the value of $K - d(i,j)$ is mapped onto $[0,1]$ (by subtracting the minimum such value, and dividing by the maximum minus the minimum).

We experimented with the incorporation of magnitudes into this "effect-of-j-on-i" term, leading for example to $w_j/d^2(i,j)$ where w_j was the magnitude of star j. An alternative scheme is to use $\mid w_i - w_j \mid .(K - d(i,j))$. We currently recommend against doing this, since there can be appreciable differences in the distributions of the values of the two terms (i.e. $\mid w_i - w_j \mid$ and $K - d(i,j)$, for all pairs i,j). Consequently one or other of these terms can predominate. Standardization or normalization, per se, does not allow us to bypass this difficulty.

To summarize: with each point i in a given point-list containing in total n points, we now have have a set of "effect" terms induced by the remaining $n-1$ points. These "effect" terms, $\{p_{ij} \mid j = 1, 2, \ldots, n;\ j \neq i\}$ have value $p_{ij} = K - d(i, j)$. Furthermore, this set of "effect" terms is ordered by the angle between j and an arbitrary axis through i. Without loss of generality, this arbitrary axis may be taken as parallel to the x-axis of the given coordinate values, and the ordering may be determined in a counter-clockwise fashion. The "world view" of point i is thus expressed by this ordered set of $n - 1$ values. Matching will later be carried out by seeking a point, or points, in the second list with a similar "world view".

The "world view" list of i may be interpreted as the set of projections (defined in a particular way) of $n - 1$ points onto the unit circle of center i.

Any "world view" in point set A is an ordered $(n-1)$-list. Any "world view" in point set B is an ordered $(m - 1)$-list. Optimally matching vectors of differing lengths can be carried out using dynamic programming. See Kruskal (1983), Sankoff and Kruskal (1983), or Hall and Dowling (1980), for discussion and practical examples. We adopted a different approach. Given that 2-dimensional data is under consideration, the angles necessarily lie between 0^o and 360^o. Hence, instead of using the given ordered list, we map this into a list of length 360 corresponding to the "world view" of a point sampled at 1^o intervals. The choice of 1^o intervals is quite arbitrary. It was found to offer a good compromise between sensitivity and computational cost. Experiments with of the order of 20 to 70 points in two lists, using 10^o intervals, were also successful. To determine the "world view" at a given angle (at an interval of 1^o from the previous and subsequent angles), we interpolated from values of p at the angles which were larger and smaller. Note that we must allow for the fact that angles mod 360 are used. Linear interpolation was used: it was simple to implement, and gave satisfactory results.

Rebinning the "world view" vectors in this manner was carried out to allow the use of the usual Euclidean distance between the new (360-valued) vectors. There are pitfalls here: *linear* rebinning does not necessarily take a continuous "world view" into account; rebinning to 1^o intervals may not be appropriate for dense point-sets; more awkwardly such bin sizes may well be problematic for closely-packed points in a given point-set. Although the simple approach adopted worked well, it is clear that further study of these issues could be profitable.

One aspect of the linear interpolation to 1^o intervals will be commented upon. Points which are outlying have a "world view" which is entirely encompassed within a limited range of angles. This angle interval can be small (e.g. considerably less than 45^o: cf. points towards the four corners of the point-sets shown in the Figures below). We found it unproductive to determine (interpolated) p values outside of this angle interval. Hence we did not approach the interpolation on the basis of a sequence of angles with 1^o separations, determining p values on each occasion. Rather, we took the initially given set of angles associated with any point's ordered "world view" list; and *interpolated* at the 1^o-separation angles which were covered or spanned by this.

As is clear from the foregoing, particular implementation choices were made in a number of instances. The solution proffered, thus far, can be stated as follows.

- The effect of j on i is given by $K - d(i,j)$. The associated angle is θ_{ij}. The "world view" of i is the set $\{K - d(i,j) \mid 1 \leq j \leq n-1\}$ which is ordered by increasing value of $\{\theta_{ij} \mid 1 \leq j \leq n-1\}$.

- Each such $(n-1)$-valued "world view" vector is mapped (by linear interpolation) onto a new 360-valued "world view" vector.

The "world view" of any point, expressed as an ordered 360-list, is now directly comparable irrespective of what point-set the point came from. The (360-valued) "world view" vector of points is now compared to the "world view" vectors of all points in the second point-set. The usual Euclidean distance is used. Point i is *matched* with a point from the second point-set when the corresponding "world view" vectors have minimum Euclidean distance.

This minimum Euclidean distance can be used as a measure of how good the match is, since it represents how similar the "world views" are. To facilitate interpretation, such match values are discretized to a [1, 10] confidence scale. This allows the results of the matching to be expressed as: "Point i from the first list is mapped onto point i' from the second list, with confidence 4", for example. Only matches above some user-specified confidence threshold, which correspond to small distances between "world view" profiles, are used to determine an expression for the overall transformation between A and B. A threshold confidence of 3 (i.e. confidences $= 1, 2$, or 3) was found to perform well.

Even within these high-confidence matches, there can be discrepancies. An average high-confidence transformation could be determined, which takes A into B. We instead favored a robust estimate, and found the median of these high-confidence values to provide satisfactory results.

Rotation is incorporated into this algorithm as follows. We consider all possible matchings between A, and 360 versions of B: i.e. the "world view" vectors of B would be all together rotated by 1^o in successive versions. We would seek a best matching from the 360 results. Computationally, this implies 360 runs of the above algorithm. If the user knows the approximate angle of rotation, then some restricted angle-interval, alone, can be searched. We have experimented with all 360 rotations of point-set B vis à vis point-set A, and also restricted (e.g. 10 degree) intervals, and results were quite conclusive in all cases. The enhanced algorithm to handle rotation is as follows:

- For a given point in point-set A, and for each permitted rotation-angle (by convention, point-set B is rotated), determine the best matching point in point-set B. Store the following: the given point in point-set A; the matched point in point set B; and the rotation-angle of point set B with which this match is associated.

- Define the appropriate rotation-angle for point set B as that angle for which the majority of best matches were found.

In our experimentation, we have generally found 80–90% of matches to indicate a unique rotation-angle. A lower threshold of, e.g., 30% is currently used to signal a lack-of-consensus situation, and hence unmatchable point-sets.

As currently implemented, we have not catered for "flipping" of points, i.e. reflection in an arbitrary axis. A solution to this could be based on reversing the order of the "world view" vector values of one of the point-sets.

If $O(n)$ points are provided in either point-set, our approach requires $O(n^2)$ time to determine the "world views" of all points; and subsequently $O(n^2)$ time to carry out the matching. Storage is seen to be $O(n^2)$.

4 Results

Figures 1 and 2 show the result of two reductions of images of the open cluster M11, obtained and studied by P.B. Stetson using DAOPHOT in 1985. Note, for example, how 28, 26 and 43, 61, 58, 24 in the first point-set ought to map onto 27, 25 and 36, 57, 52, 23. Note that 38, 20, 45, 14 in the upper left-hand side of the first point-set are absent in the second point-set. The matching obtained by Stetson's triangle-based matching approach (discussed in Section 2 above) is shown in Table 1.

Figs. 1 and 2; Tables 1 and 2; Figs. 3 and 4; Table 3, in this order, near here.

The results obtained by the approach described in this paper are shown in Table 2. Note that not all correspondences between points in the two lists are well matched; but that an acceptable subset of points are. The mapping of 9 onto 46, for instance, is correctly down-graded in confidence by our algorithm. Note how 5, 16, 2, 23, 56 from the first point-set (Figure 1) are correctly mapped with high confidence onto 5, 16, 2, 26, 51 (respectively) from the second point-set (Figure 2). A sufficient number of high-confidence correspondences suffices to define the appropriate mapping which takes the first point-set onto the second.

The exact value of our translation vector, taking the first point-set into the second point-set, differs in precision from that yielded by P. Stetson's algorithm (cf. these translations as given towards the ends of Tables 1 and 2). Note that both translation values are a result of particular definitions (ours is a particular median value). Furthermore the precise definitions of both translations are not inherently coupled to the matching algorithm, and could be replaced by alternative definitions.

Figures 3 and 4 were derived from a figure in Groth (1986). A digitized photographic image provided the stars in one set, and the points in the other set were culled from a catalog. Eighteen points in Figure 3 correspond exactly with points in Figure 4. The results obtained are shown in Table 3. Note how point 1 from the first point set (Figure 3) is mapped correctly onto point 1 in the second point set (Figure 4), but with relatively unfavorable confidence. The "world views" of 1 in the two point-sets are clearly confused by neighboring points. Note that all high-confidence matches between these two point sets (i.e. matches with confidence = $1, 2$, or 3) are correct. We only seek a number of such high-quality matches in order to define the relationship between the two point-sets.

Figs. 5 and 6; Figs. 7, 8 and 9; Tables 4 and 5, in this order, near here.

Figures 5 and 6 show two point-sets to be matched (data courtesy of S. Ortolani). A magnitude-limit of 14.0 yielded the point-sets shown in Figures 7 and 8. The magnitude of 14.0 is arbitrary, with the sole requirement that around 100 points (a compromise between many points, leading to a robust solution, versus computational expense) should result in either list. Results for the matching of the latter two point-sets is shown in Table 4.

Figure 8 was rotated by 25^o clockwise: see Figure 9. A matching between Figures 7 and 9 therefore used the potential of our algorithm for handling rotation. One result, related to user-constraining of what rotation angles were to be searched, is shown in Table 5.

Using the transformation yielded by the feature-based algorithm on magnitude-limited point-sets, the full matching of all points shown in Figures 5 and 6 was carried out. We use a rough measure of acceptable correspondences as a matched distance of less than 1.0. Using this measure, we find that 1685 points are matched from 1883 points in Figure 5 and 2552 points in Figure 6.

Sample timings of the method implemented are as follows. Feature-based matching for about 100 points in both point-sets, without rotation, requires about 25 seconds CPU time on a SPARCstation 2. For rotation, about 14 seconds per degree checked out is required. Given the rotation angle and the translation, a full matching of the data shown in Figures 5 and 6 (comprising about 2000 points in the two point sets) takes about 18 seconds CPU time on a SPARCstation 2. In this latter case, a brute-fore, unintelligent best match algorithm (i.e. $O(n^2)$) was implemented.

A range of cleverer approaches for best match searching in two dimensions (some of which are reviewed in chapter 2 of Murtagh 1985) would considerably speed up this phase of the processing. In fact, it is well known that (perhaps surprisingly) a nearest neighbor can be obtained in *constant* expected time (i.e., independent of the sizes of the point sets: see Bentley et al. 1980). It is also probably the case that the efficiency of the feature-based phase of the processing could be studied, and speed-ups affected.

5 Discussion

We have presented an efficient algorithm for point pattern matching, and demonstrated its success in handling invariance of the following types: translation, scaling, perturbation, random insertions and deletions, and rotation. Further enhancement of the algorithm could handle reflection in an axis.

Within the framework of the approach described, a number of possibilities for further fine-tuning have been noted. It would be interesting to investigate the relationship between the definition of a "world view" and spherical factor analysis, a little used technique which was explored in an 80-page article by Domengès and Volle (1979).

The algorithm described in this paper has considerably better computational complexity characteristics, and/or applicability properties, compared to algorithms which are currently in use as auxiliary tools in the area of 2-dimensional photometry. The procedure is robust, in terms of positional coordinates, and in terms of magnitude (when this is used). Specific breakdown points have yet to be investigated.

Triangle-based mapping (the work of Groth and Stetson discussed above) is based on differing presuppositions to those used in this article. The approach we have described has been found to achieve a matching of adequate quality in an efficient and robust manner.

Acknowledgements

Motivation for the approach described here arose from discussions with the authors, whose data sets are used (with thanks) in the Figures. I am also grateful to an anonymous referee for suggesting various improvements in the paper.

References

Bentley, J.L., Weide, B.W., & Yao, A.C. 1980, ACM Trans. Mathematical Software, 6, 563

Chen, H.H., & Huang, T.S. 1991, IEEE Trans. Pattern Analysis and Machine Intelligence, 13, 872

Domengès, D., & Volle, M. 1979, Annales de l'INSEE, No. 35, 3

Griffin, P.M., & Alexopoulos, C. 1989, IEEE Trans. Systems, Man, and Cybernetics, 19, 1274

Groth, E.J. 1986, AJ, 91, 1244

Hall, P.A.V., & Dowling, G.R. 1980, Computing Surveys, 12, 381

Johnston, M.D., & Adorf, H.-M. 1991, J. Computers and Operations Research, in press

Kruskal, J.B. 1983, SIAM Review, 25, 201

Murtagh, F. 1985, Multidimensional Clustering Algorithms (Wuerzburg, Physica-Verlag)

Nasrabadi, N.M., & Choo, C.Y. 1992, IEEE Trans. Neural Networks, 3, 5

Ogawa, H. 1984, Pattern Recognition, 17, 569

Ogawa, H. 1986, Pattern Recognition, 19, 35

Parvin, B., & Medioni, G. 1989, Proc. International Joint Conference on Neural Networks, vol. II, 281

Sankoff, D., & Kruskal, J.B. 1983, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison (New York, Addison-Wesley)

Stetson, P.B. 1990, The techniques of least squares and stellar photometry with CCDs, Dominion Astrophysical Observatory preprint

Wong, A.K.C., & Salay, R. 1986, Proc. Eighth International Conference on Pattern Recognition, Paris, France, Vol. 1 (New York, IEEE Computer Society Press), p. 546

Umeyama, S. 1991, IEEE Trans. Pattern Analysis and Machine Intelligence, 13, 376

Figure Captions

Figure 1: A point-set, to be compared with Figure 2.

Figure 2: A point-set, to be compared with Figure 1.

Figure 3: A point-set, to be compared with Figure 4.

Figure 4: A point-set, to be compared with Figure 3.

Figure 5: A point-set, to be compared with Figure 6.

Figure 6: A point-set, to be compared with Figure 5.

Figure 7: A magnitude-limited subset of the point set in Figure 5; to be compared with Figure 8.

Figure 8: A magnitude-limited subset of the point set in Figure 6; to be compared with Figure 7.

Figure 9: A -25^{o}-rotated version of Figure 8; to be compared with Figure 7.

Table Captions

Table 1: Result of matching using P.B. Stetson's routine, on data shown in Figures 1 and 2.

Table 2: Results of feature-based algorithm on point-lists shown in Figures 1 and 2.

Table 3: Results of feature-based algorithm on point-sets shown in Figures 3 and 4.

Table 4: Results of feature-based algorithm (first 10 points only) on point-sets shown in Figures 7 and 8.

Table 5: Results of feature-based algorithm (first 10 points only) on point-sets shown in Figures 7 and 9.