HAL open science

Pattern Matching with Differential Voting and Median Transformation Derivation

Marcin Marszalek, Przemyslaw Rokita

To cite this version:

Marcin Marszalek, Przemyslaw Rokita. Pattern Matching with Differential Voting and Median Transformation Derivation. International Conference Computer Vision and Graphics (ICCVG '04), Sep 2004, Warsaw, Poland. pp.1002–1007, 10.1007/1-4020-4179-9_146 . inria-00548539

HAL Id: inria-00548539

https://inria.hal.science/inria-00548539v1

Submitted on 20 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PATTERN MATCHING WITH DIFFERENTIAL VOTING AND MEDIAN TRANSFORMATION DERIVATION

Improved Point-Pattern Matching Algorithm for Two-Dimensional Coordinate Lists

Marcin Marszałek[1] and Przemysław Rokita[2]
Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
[1]mmarszal@elka.pw.edu.pl; [2]pro@ii.pw.edu.pl

Abstract We describe an algorithm for matching two-dimensional coordinate lists. Such matching should be immune to translation, rotation, scaling and flipping. We assume that coordinate lists may be only partially overlapping and we allow some random perturbations in the lists. Our goal is to match enough points to be able to derive a coordinates transformation with high confidence. The implementation described here is a part of SkySpy automated sky survey system. The presented algorithm is used to perform matching of stars detected during observations with star catalogues.

Keywords: pattern matching; coordinate lists; differential voting; transformation derivation

1. INTRODUCTION

One of the tasks an automated sky survey system must perform is matching stars detected on acquired CCD frames with star catalogues. After astrometry (position measurement) in frame coordinates and relative photometry[7] (brightness measurement) has been performed, we are given a list of observed stars positions and brightnesses. On the other side a list of stars contained in the star catalogue[3] is available, where for each star we are given its position in celestial coordinates and brightness in magnitudo scale.

As new frames are continuously acquired, implementation of a fast and robust algorithm, capable of directly matching tens of stars found on a frame with millions of stars contained in a catalogue, accordingly to the literature seems to be impossible nowadays. The solution we have found to fulfill the described requirements is to derive a coordinates transformation and use a simple and fast algorithm to match close neighbors in the same coordinate space.

To derive a coordinates transformation between frame coordinates and celestial (catalogue) coordinates, we still need to perform pattern matching[5],

but we can focus on matching only few stars, enough to derive a transformation. Those stars should be matched with high confidence.

To reduce the number of considered catalogue stars, we use some additional information (measured star movement, observer location, hardware parameters) to calculate a rough estimate of observed sky region. Later on, previously found transformation parameters can be used as the hints for the algorithm.

Although simplified, pattern matching in our case is still not a trivial task. The lists are not only subject to translation, rotation, scaling and flipping, but the areas concerned may be only partially overlapping, there may be random star additions and deletions (due to existence of non-star objects and measurement errors caused by cosmic rays or clouds for example) and random coordinates and/or brightnesses perturbations. Moreover, after pattern matching the list of paired points can contain occasional mismatches. We need an algorithm to quickly determine transformation parameters that would not be heavily affected by occasional mismatches, which is not a trivial task as well.

In the following sections we will describe the algorithms used to perform the pattern matching and coordinates transformation derivation. We will shortly present previous achievements and describe improvements we have developed.

2. OVERVIEW

As an input for our pattern matching algorithm we are given two lists of two-dimensional coordinates and measured object brightnesses. We need to derive a coordinates transformation between the two lists.

Many algorithms were proposed to solve the stated problem (see the research of Murtagh[6]). We have based our work on the solution presented by Groth[2] and developed in parallel (although not published) by Stetson[8]. This approach, mimicking the way an astronomer performs matching sky images, was later improved by Valdes[10] and tested by Hroch[4].

The idea of the algorithm is to search for similar triangles constructed from the points on the two lists. We expect the relating points in the lists to generate many similar triangles, but also expect occasional matches generated by non-corresponding objects. To derive a list of objects corresponding with high confidence, we propose a differential voting method, which we believe is an important improvement compared to previously proposed method. Having a list of relating objects we derive a transformation between the two lists. As at this step a dominant majority of points are true matches, we propose an efficient method of median derivation of transformation parameters.

As we are dealing with spherical coordinates, we use the direction cosines and matrix method for coordinates transformation where applicable. This method for solving celestial mechanics problems was proposed by Taki[9].

3. THE ALGORITHM

First, we choose the m brightest objects from the first list and n brightest objects from the second list. In fact, all previous works assumed the same

number of elements in matched lists, though the algorithm does not require this assumption. In our case it is better to choose more objects from the catalogue list. Having only rough estimate of the area covered by the frame, we intentionally overestimate this parameter and pick more objects from the catalogue. We expect the best results, when a number of relating objects in two lists is maximized, so we try to provide a similar densities of chosen objects over the area. This implies selecting more objects from the catalogue list.

We select the brightest objects, as those can be found on a CCD image with high confidence and we can expect the brightest objects in one list correspond to the brightest objects in the second list. It is worth noting, that we do not require the same brightness ordering for the stars in the two lists. We have found that selecting 15 to 30 brightest objects is sufficient for the algorithm, giving good results with not too long computation times.

3.1 Finding similar triangles

Provided the reduced object lists, our next step is to construct all possible triangles. From the list of n objects we can construct $n(n-1)(n-2)/6$ triangles. Those will be represented in so called "triangle space".

We need to choose a triangle representation that will let us find similar triangles, that is to find corresponding object triplets being insensitive to translation, rotation, scaling and flipping. We find a representation proposed by Valdes better than the one originally presented by Groth. A triangle in the triangle space could be represented as a two-dimensional point (x, y) where

$$x = a/b, \quad y = b/c$$

a, b and c are the lengths of triangle sides in decreasing order. Similar triangles will be located close to each other in the triangle space.

Two techniques that allow to speed up a triangle generation and search are worth noting here. First, all the distances between objects in reduced list may be precalculated (optimization introduced by Valdes). This prevents calculating the distance between same objects many times when lengths of triangle sides are determined. Second, the triangles in the triangle space may be presorted by one of the coordinates (optimization introduced by Groth). Then we use a moving window over the presorted coordinates to heavily reduce the number of candidates for neighbors for each triangle considered. A similar technique based on presorting the triangles in the triangle space was proposed by Valdes, but we have found that as we wish to find all similar triangles anyway, the binary search to locate the subset is not necessary.

Also, as we were going to use differential voting later, we did not follow Groth's techniques of filtering out the non-useful triangles, nor a simplified approach presented by Valdes. By using our new approach, the triangles generating many false matches can easily be discarded in differential voting process.

It is worth noting (as it wasn't directly stated in previous works), that the presented pattern matching algorithm should be applicable for any two-dimensional

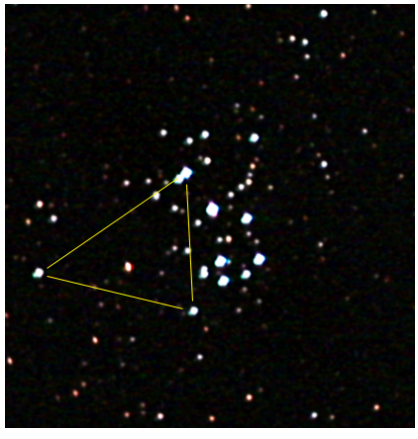

Figure 1a. CCD image of M45 (Pleiades) open cluster. Three stars creating an example triangle are marked.

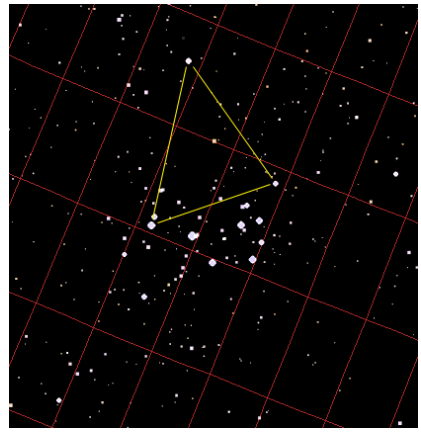

Figure 1b. Catalogue data for M45 open cluster. Corresponding triangle is marked. Celestial coordinates grid is shown.

surface with a natural geometry. We use it for spherical geometry, but its use for Euclidean or hyperbolic geometries seems to be evident.

3.2 Differential Voting

When similar triangles have been found, we use voting method to find corresponding objects. Each matching triangle pair votes for all of its vertices pairs. The argumentation for this technique is based on the fact, that corresponding vertices pairs will receive votes from many similar triangles, while non-corresponding pairs will receive few votes from occasional mismatches.

An important improvement of the algorithm we have found is to perform differential voting, that is to treat as corresponding not the pairs that received the largest number of votes, but the pairs that received the largest number of votes comparing to other pairs with the same member.

The reasoning for our improvement is that we expect one-to-one matching of objects in the two lists. As we require an object matching with high confidence, if an object receives many votes in two pairs (with two different elements), we do not want to trust neither of the associations.

We propose to subtract from the number of votes, received by an object pair in traditional way, the number of votes received by the pair with the highest number of votes and same member. That is

$$v'(m,n) = v(m,n) - \max(\max_{k \neq m}(v(k,n)), \max_{l \neq n}(v(m,l)))$$

where $k, m \in M$, $l, n \in N$, $v(m,n)$ is the number of votes pair (m,n) has received in a traditional voting and $v'(m,n)$ is a number of votes received in a proposed approach.

Table 1a. Vote array before correction. Six highest vote getters were marked bold. Note the obvious mismatch in the upper-left corner.

17	5	2	1	0	0	3	2
16	6	2	1	0	0	4	2
6	15	1	1	2	0	5	3
0	1	**20**	5	2	0	1	9
0	0	4	5	2	2	0	2
0	0	8	3	3	**31**	0	10
0	6	2	0	**22**	1	3	5
1	3	0	1	0	2	15	4
0	2	13	1	4	2	3	**22**
1	2	1	1	6	1	3	2

Table 1b. Vote array after correction. Six highest vote getters were marked bold. Pairs that received negative number of votes were zeroed.

1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	**9**	0	0	0	0	0	0
0	0	**7**	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	**21**	0	0
0	0	0	0	**16**	0	0	0
0	0	0	0	0	0	**10**	0
0	0	0	0	0	0	0	**9**
0	0	0	0	0	0	0	0

Note, that for each object in any of the lists, there will be at most one object on the other list, that forms a pair that has received a positive number of votes. That is for each object there will be at most one matched object, and the number of votes will be proportional to the confidence of one-to-one matching.

3.3 Coordinates transformation derivation

When we have object pairs that can be associated with high confidence, we want to derive a coordinates transformation between two lists. First we want to know if the frame is flipped and we need to correct the rough estimate of the field of view (this is scale the frame in celestial coordinates). Then we derive the celestial coordinates of frame center and compute the frame rotation.

We can determine whether the frame is flipped by constructing triangles[1] from matched object pairs (from few highest vote getters in fact) and perform voting. At the same time we can calculate the scale correction factor from doubles of matched object pairs. We propose not to calculate the average of the factor, but compute the median value, as occasional mismatches still may occur.

Having corrected the orientation and the scale we can derive the remaining spherical coordinates transformation parameters using the direction cosines and matrix method for coordinates transformation derivation as proposed by Taki[9]. Again, we calculate the final value of each transformation parameter as median of the values derived from all doubles of pairs. We have found that this is a very simple and robust way to minimize the influence of the rare mismatches left after the pattern matching phase.

Our tests showed, that after differential voting phase, when we have chosen six most voted object pairs to derive the transformation parameters, more than 80 percent of results were correct.

For other, especially non time-critical applications, an iterative sigma clipping algorithm proposed by Valdes[10] may be used.

4. SUMMARY

The presented algorithm allows our automated sky survey system to find a coordinates transformation between CCD frame coordinates and celestial coordinates. The triangle matching and voting procedure successfully pairs the objects regardless of minor perturbations in the coordinate lists, as far as there are enough (six or more) bright objects appearing on both lists.

A differential voting method discards most of the mismatches and a coordinates transformation parameters may quickly be found. Our approach allowed us to abandon the complicated filtering methods and still reduce the number of mismatches, especially among highest vote getters.

Before we have implemented the differential voting correction, mismatches appeared even among the three highest vote getters. After applying the correction, we can use seven highest vote getters and mismatches among those are very rare. In this situation median transformation derivation can be used as quick and robust way to derive transformation parameters. As the proposed voting correction may be applied to original voting table with no computational penalties (the algorithm is $O(mn)$), we believe our finding may be useful.

On a 1.1 GHz PC computer the average computation times for matching two point sets of 30 elements each, were 0.03s for triangle space construction phase and 0.7s for fiding corresponding pairs. Introducing our differential voting correction added 0.1s to pair matching phase.

References

[1] I.N. Bronstein and K.A. Semendjajew. *Handbook of Mathematics*. Springer-Berlin, 2004.

[2] E.J. Groth. A Pattern-Matching Algorithm for Two-Dimensional Coordinate Lists. *Astronomical Journal*, 91:1244–1248, May 1986.

[3] E. Hfig et al. The Tycho-2 Catalogue of the 2.5 Million Brightest Stars. *Astronomy and Astophysics*, 355:27, January 2000.

[4] Filip Hroch. *Computer Programs for CCD Photometry*. Department of Theoretical Physics and Astrophysics, Masaryk University, Brno. http://munipack.astronomy.cz/doc/munipack.html.

[5] R. Klette, K. Schluns, and A. Koschan. *Computer Vision*. Springer, 1998.

[6] F. Murtagh. FOCAS Automatic Catalog Matching Algorithms. *Publications of the Astronomical Society of the Pacific*, 104:301, 1992.

[7] J. Palmer and A.C. Davenhall. *The CCD Photometric Calibration Cookbook*. Council for the Central Laboratory of the Research Councils, 2001.

[8] P.B. Stetson. *The Techniques of Least Squares and Stellar Photometry with CCDs*. Dominion Astrophysical Observatory. http://nedwww.ipac.caltech.edu/level5/Stetson/frames.html.

[9] Toshimi Taki. *Matrix Method fot Coordinates Transformation*, 2002. http://www.asahi-net.or.jp/ zs3t-tk/matrix/matrix.htm.

[10] F. Valdes et al. FOCAS Automatic Catalog Matching Algorithms. *Publications of the Astronomical Society of the Pacific*, 107:1119, November 1995.