Matching Images With Quickmatch

Robert Quimby
May 30, 2002

1. Overview

Quickmatch is a package designed to identify a set of matched objects from two images. The package consists of an IDL routine that reads in raw images and produces a clean astrometry list (GET_PHO.pro), a C routine which takes the two lists and finds a matched set of objects within some scale and rotation constraints (QUICKMATCH.c), and an IDL wrapper which calls these routines given a pair of image names (QUICKMATCH.pro).

2. Setting the Match Parameters

The match parameters lend some flexibility to the user to find a compromise between speed and accuracy. Adjusting one parameter can have important consequences to others, so users must understand how all the parameters function and work together before attempting to tweak values from their defaults.

Parameter Name	Default	Description	Speed Impact
n_needed_verify	5	Must have at least this number of matching nodes to verify a match	none
percent_good_XXX	0.5	Percentage of objects in the over lap that are expected to match	little
dist_tol_XXX	2.5	Distance tolerance for a match.	yes
pix_per_100_XXX	0.4	The dist_tol is increased by this amount for every 100 pixels a neighbor is distant (capped at 1.6*dist_tol; dist_tol*=1.2)	little
fail_rate	0.01	Sort of the statistical fail rate	large
n_scales	1	Number of discrete scales to check for	$0(n)$
scales	[1.00]	An array containing the allowed image scales (new/ref)	N/A
scale_margin	0.90	Scale tolerance. Matches are allowed from scales[i]*scale_margin to scales[i]/scale_margin	huge
n_phases	2	Number of discrete rotations to check for	$0(n)$
phases	[0,180]	An array containing the allowed image rotations (new-ref) in degrees	N/A
phase_margin	90.0	Phase tolerance in degrees. Matches are allowed from phases[i]-phase_margin to phases[i]+phase_margin	mild
do_flip	0	Set to 1 to enable parity (flip) checking.	$0(n)$
time_allowed	10	Time allowed to find a match in seconds.	none
min_pix_XXX	5	Minimum number of pixels in an object. Used to reject cosmic ray like events.	little
thresh1	2.0	Passed to ISOFIND . Pixels must be thresh1 standard deviations above the mean to be included in an object.	little
thresh2	3.0	Passed to ISOFIND . Objects must have at least one pixel above thresh2 STD.	little

3. Preparing the Photometry Lists

Any matching algorithm must be fed clean lists of objects in order to function properly. It is important that the lists contain all of the objects in the overlap, with as few false detections as possible. Bad columns can be identified by performing histograms along both the axes. Also, the lists can be searched for over-densities created along boarders and near saturated objects.

Lists can be created automatically, or with user input using GET_PHO .

4. Matching

Given two lists of objects, QUICKMATCH will attempt to match objects in the lists with in the scale and rotation constrains passed in the match parameters. The first step is to create nodes at each object with the distance and angle to all of the other objects in the list.

To match, QUICKMATCH first employs an algorithm similar to that developed for the SCP. For each node in the first list it searches through each node in the second and compares 2 neighbors from the first, selected at random from a set of 15, to the neighbors in the second node. The 15 neighbors are chosen such that for the first objects in the list (presumably the brightest), the first 15 in the list are considered (again, presumably the 15 brightest), and for the last nodes in the list, the 15 closest neighbors are used. The 2 neighbors from the first node are compared with those in the second to see if a pair results in an allowed scale and rotation. If successful, the calculated scale and rotation is applied to all 15 neighbors, and the total number of neighbor matches are counted.

A match is signaled for a pair of neighbors if the error ellipses created by the distance and angle tolerances overlap when transformed to the same space. The distance tolerance is usually taken to be the image seeing, plus an amount to compensate for the lever arm effect of the over all distance:

dis_tol=1.2*((pix_per_100*r/100 < dist_tol) ? (dist_tol+pix_per_100*r/100.0)/2.0 : dist_tol);

In cases where this first method fails to produce a match, a second algorithm is tried which is much more thorough at the cost of overall speed. Instead of just using 2 of 15 neighbors from the first node, 2 or more neighbors are selected from the entire set. The exact number to try is determined by the fail_rate specified (see get_n_needed). This procedure is much better suited to matching short object lists, fields with small over lap, and lists produced under different filters such that the ordering of objects by flux will be substantially different.

Algorithm	Speed	List Requirements
quickmatch_pattern	$0(n_1\, n_2 \log(n_2)])$	Ordered by flux
match_pattern	$0(n_1^3\, n_2[\log(n_2)]^2)$	none

5. Match Verification

The final step in the matching process is to take the few objects matched, calculate a new scale and rotation, and to see how many objects in the overlap match. The overlap region is calculated by assuming rectangular fields, and using the objects at the extremes to mark the boundaries. The number matching objects needed to verify the match is calculated by considering the number of objects on each list in the overlap, and the percent_good_XXX values. Multiplying and taking the smaller number gives the minimum number of matching objects to verify a match (the percent_good_XXX values are increased in cases of small overlap, and the minimum number is not allowed to fall below n_needed_verify).

6. Test Results

The algorithm was tested (using only match_pattern) by creating photometry lists with GET_PHO from images taken of supernovae discovered by NEAT for the Spring '99 Nearby Supernovae Search. The results show the algorithm to function as designed. The only images that it failed to match are either long exposure U band images with very few true objects buried in a sea of cosmic rays, or images that appear to have had read errors, or other catastrophic problems. All other images are matched.

The following tables relate the success of the procedure. The tests were performed using pre-determined scales and tight scale constraints. The tests were performed on all images with overlap (according to the headder information), regardless of image quality. Images in the "FAIL" column were taken in a filter other than the U band, and did not produce a match for reasons that are understood; namely, the 10 Danish images that failed were multiple exposure focus images, while the 54 failed Lick images were long exposures, primarily in the B band taken through clouds producing very few objects, but many cosmic rays.

REFNAME	MATCHED	NO MATCH	PERCENT
mar2498neatb706com.fts	99	3	97.058
mar2498neatb720com.fts	99	3	97.058
feb1899neatb103324csg.fts	100	2	98.039
feb1899neatb104855csg.fts	100	2	98.039
feb1899neatb111359csg.fts	100	2	98.039
mar2498neatb652com.fts	98	3	97.029
mar2498neatb907com.fts	65	12	84.415
mar2498neata919com.fts	67	10	87.013
mar2498neatb921com.fts	65	12	84.415
mar2498neata933com.fts	67	10	87.013
mar2498neatb936com.fts	65	12	84.415
mar2498neata905com.fts	67	9	88.157
mar2598neatb701com.fts	160	15	91.428
mar2598neatb715com.fts	164	11	93.714
mar2598neatb730com.fts	165	10	94.285
feb1999neatb101226csg.fts	163	12	93.142
feb1999neatb102715csg.fts	151	24	86.285
mar2698neatb1137com.fts	35	0	100.000
mar2698neatb1152com.fts	35	0	100.000
mar2698neatb1123com.fts	34	0	100.000
mar2598neata625com.fts	102	6	94.444
mar2598neata639com.fts	102	5	95.327
mar2598neata610com.fts	101	6	94.392

REFDET	MATCHED	NO MATCH	FAIL	PERCENT
neatb	1698	123	48	93.245
neata	506	46	16	91.666

NEWDET	MATCHED	NO MATCH	FAIL	PERCENT
yalo	321	37	0	89.664
li1m	423	101	54	80.725
ctpn	146	0	0	100.000
kp2m	279	0	0	100.000
ct1p5m	526	0	0	100.000
dan	413	19	10	95.601
cfht	81	0	0	100.000
ml40	9	12	0	42.857
jkt	6	0	0	100.000

MATCHED 2204

COULD NOT MATCH 169

MATCH PERCENT (FULL SET) IS 92.8782

MATCH PERCENT (NOT LICK, NOT U-BAND) IS 99.3506

MEDIAN TIME IN C ROUTINE WHEN MATCHED 0.229700 seconds

MEDIAN TIME IN C ROUTINE WHEN _NOT_ MATCHED 1.29125 seconds

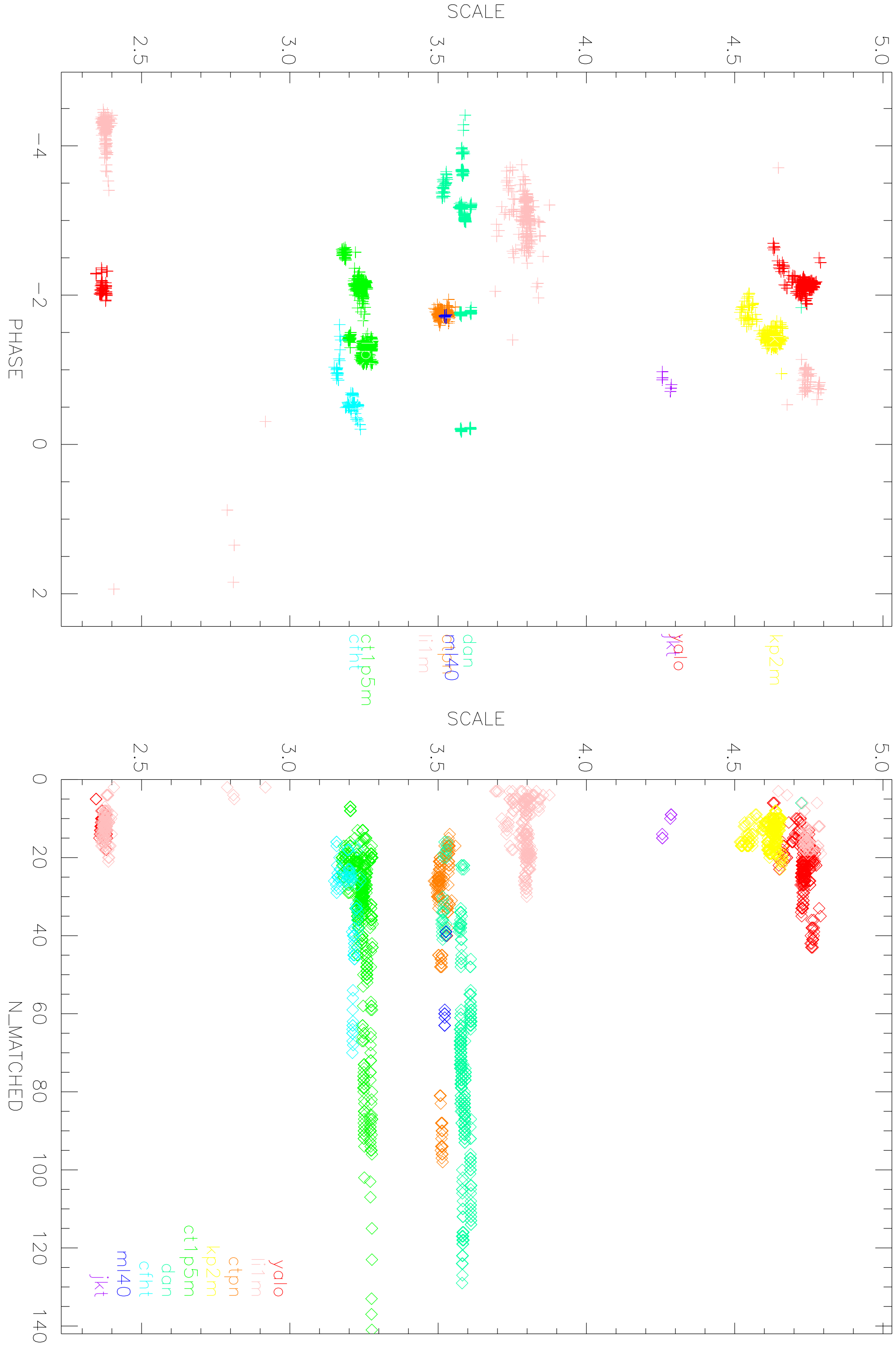

Fig. 1.— Test results. These plots can be used to ascertain the validity of a match. Matches should cluster in scale and phase space.

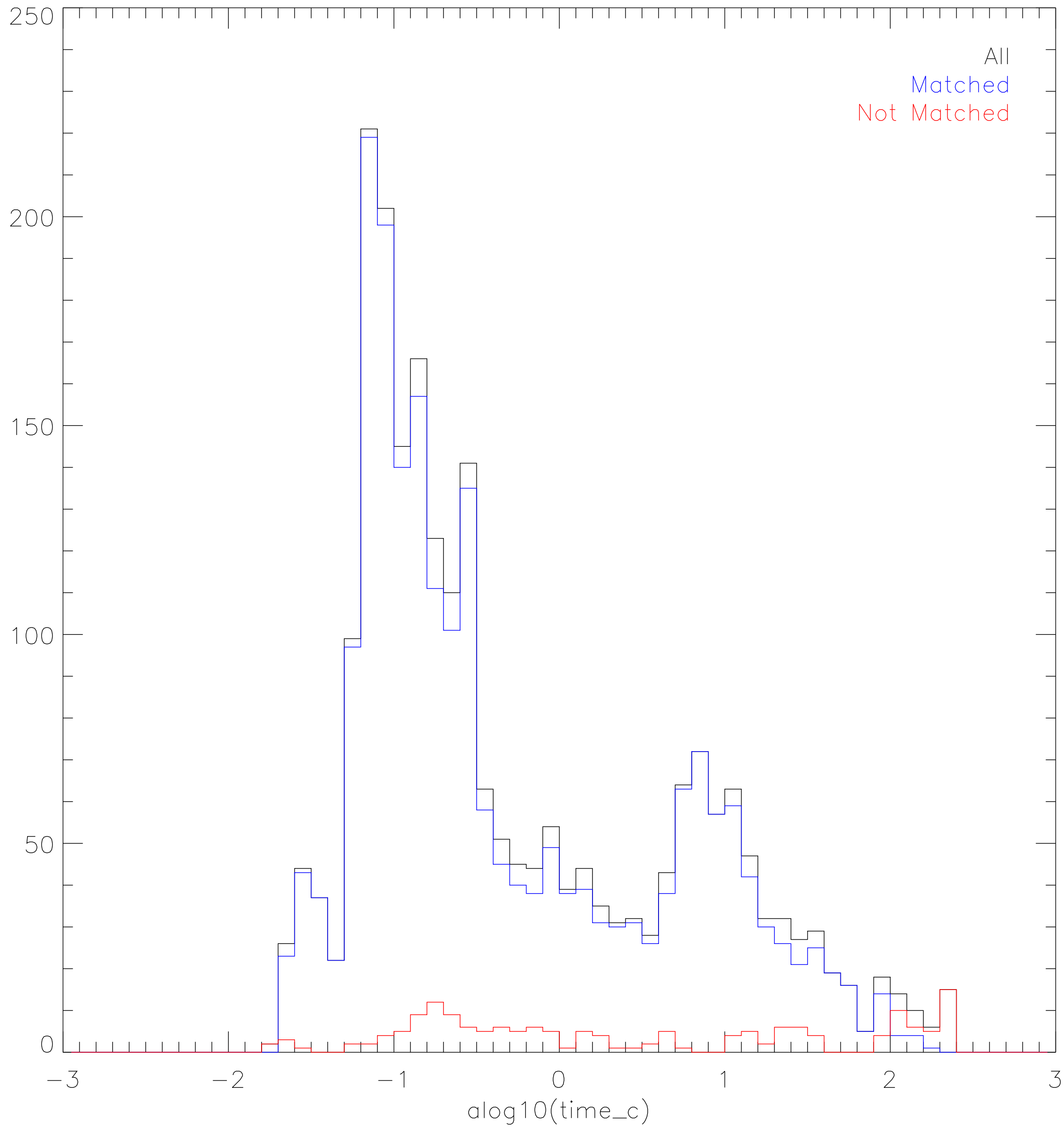

Fig. 2.— Time spent in the C routine for all cases tested. An upper limit of 300 seconds was set.