Mon. Not. R. Astron. Soc. **000**, 1–11 (2003) Printed 4 September 2003 (MN LaTeX style file v2.2)

A highly-automated moving object detection package

J-M. Petit[1]*, M. Holman[2], H. Scholl[3],
J. Kavelaars[4] and B. Gladman[3,5]
[1] *Observatoire de Besançon, BP 1615, 25010 Besançon cedex, France*
[2] *Harvard-Smithsonian Center for Astrophysics, MS-18, 60 Garden Str, Cambridge, MA 02138, USA*
[3] *Observatoire de la Côte d'Azur, BP 4229, 06304 Nice cedex 04, France*
[4] *National Research Council Herzberg Institute of Astrophysics, 5071 West Saanich Road, Victoria, B.C. V9E 2E7, Canada*
[5] *Currently: Dept. of Physics and Astronomy, University of British Columbia, 6224 Agricultural Road, Vancouver, B.C. V6T 1Z1, Canada*

Submitted 2003 September 4

ABSTRACT
With the deployment of large CCD mosaic cameras and their use in large-scale surveys to discover Solar System objects, there is a need for a fast detection algorithms that can handle large data loads in a nearly automatic way. We present here an algorithm that we have developed. Our approach, by using two independent detection algorithms and combining the results, maintains high efficiency while producing low false detection rates. These properties are crucial to in order to reduce the operator time associated with searching these huge data sets. We have used this algorithm on two different mosaic data sets obtained using the CFH12K camera at CFHT. Comparing the detection efficiency and false-detection rate of each individual algorithm with the combination of both, we show that our approach decreases the false detection rate by a factor of a few hundred to a thousand, while decreasing the 'limiting magnitude' (where the detection rate drops to 50%) by only 0.1-0.3 magnitudes. The limiting magnitude is similar to that of a human operator blinking the images. Our full pipeline also characterizes the magnitude efficiency of the entire system by implanting artificial objects in the data set. The detection portion of the package is publicly available.

Key words: Solar system: general – methods: data analysis – Kuiper Belt

1 INTRODUCTION AND MOTIVATION

Over the past decade, observation of the Edgeworth-Kuiper Belt has revealed a rich and surprisingly complex dynamical structure. The Kuiper belt appears to contain at least three populations of trans-neptunian objects (TNOs): TNOs trapped in Neptune's 2:3 mean motion resonance (*Plutinos*) as well as other resonances; objects with non-resonant orbits beyond 42 AU (the *classical belt*); and objects with highly eccentric orbits that may permit intermittent close approaches with Neptune (the *scattered disk*) (Duncan and Levison 1997; Gladman *et al.*, 2001). However, due to the limited number of TNOs discovered in surveys that determined orbits for a complete set of their discoveries, several aspects of the Kuiper belt are still largely undetermined. Particularly important are the relative number of members of these populations, the eccentricity and inclination "excitation" of the belt (Petit *et al.*, 1999), the radial extension of the classical belt (Gladman *et al.*, 1998; Allen *et al.*, 2002; Trujillo and Brown 2001), and the ratio between the number of bodies in the 1:2 and in the 2:3 resonance (Jewitt *et al.*, 1998).

Studies of the origin of the dynamical and physical structure of the Kuiper Belt, and the clues this gives regarding the formation of the giant planets themselves, have progressed to the point that scientists require quantitatively accurate measurements of various parameters. As a first example, the current size distribution of the Kuiper belt provides information about the accretion processes and collisional environment of this portion of the Solar System; theoretical studies (Davis and Farinella, 1997; Farinella *et al.*, 2000; Kenyon, 2002; Stern and Colwell, 1997) now need information with greater accuracy than the observational studies (Trujillo *et al.*, 2001a; Gladman *et al.*, 2001; Larsen *et al.*, 2001) can probably provide given the uncertainties due to smallish statistical samples and potential systematic errors. Of these, the ability to determine detection efficiency as a function of magnitude will be our chief concern here.

As a second example, theoretical studies of the orbital distribution of the belt need to have characterized surveys in order to compare their models with the observational results. Many discoveries of Kuiper belt objects have been made in

* E-mail: petit@obs-besancon.fr

uncharacterized surveys. For a survey to be "fully characterized" and useful for theoretical studies, it must publish (or make otherwise publicly available) the following information for the discovery (and potentially recovery) fields: date of observations, coordinates, size of detector (and filling factor if a mosaic instrument), the efficiency curve (not just a single number or detection limit) stating what rates of motion it is valid for, and targets found (or not) in case of recovery. Most Kuiper surveys published in the last two years (Gladman *et al.*, 2001; Trujillo *et al.*, 2001a; Trujillo *et al.*, 2001b; Larsen *et al.*, 2001) provide this information. The field pointing locations are important because resonant objects come to perihelion (and thus become more visible) at preferential longitudes relative to Neptune, and so calculating the observability of these populations requires this knowledge. Note that because of varying stellar density along the ecliptic it is likely that the survey efficiency even for 'bright' Kuiper Belt objects is a not just a function of seeing.

The current surveys and those to come use CCD mosaic cameras. Searching the enormous amount of data produced by these cameras for moving objects is only practical with the help of semi-automated search programs especially tuned to detect the targets of interest. Several authors have mentioned the use of such programs: Levison and Duncan (1990), Irwin *et al.* (1995), Trujillo and Jewitt (1998), Rousselot *et al.* (1999). Some of these codes were proprietary, and therefore not available to us. More importantly, they have all relied heavily upon a final operator verification, with a high false-detection level from the code overcome by having a human operator reject false detection near the magnitude limit. Therefore it was not clear that these programs would permit data reduction in a time short enough to ensure successful recovery of all detected objects in a large-scale survey. Hence we decided to create our own software. Our main goal in doing this was to create a software suite that would be flexible enough to run in various environments, on data from different telescopes, and above all, that would run fast enough to allow near real-time reduction of the data.

The most stringent time constraint arises from the need for confirming observations of recently disccovered objects. We need to detect all new objects in our discovery frames within 24 to 48 hours to allow for these confirming observations to be performed efficiently and successfully. In fact, in the past we usually worked in near real-time at the telescope, searching fields in less than 12 hours of acquisition allowing us to re-image only those fields containing TNOs on subsequent nights, thus vastly increasing our areal coverage. For the up coming Ultra-Wide component of CFHT Legacy Survey (see http://cdsweb.u-strasbg.fr:2001/Science/CFHLS/ for a description of this survey), rapid detection means processing 32 (number of fields) $\times$ 3 (number of exposures per field) $\times$ 36 (number of chips in the camera) = 3456 images of rough size 2k $\times$ 4k pixels. The automated search program should be capable of detecting objects as faint as those detected by the eyes of a human operator, while keeping the number of false detections to as low a value as possible since otherwise the rejection of false detections when operating near the detection limit is the most time-consuming portion of a human operator's time. Our software was developed for the data from the CFH12K cameras but has been successfully used for data from other single-CCD or CCD mosaic cameras, such as at the KPNO 4-meter, Palomar 5-meter, ESO 2.2m with WFI, or the VLT FORS cameras. It is presently being used for the new MegaPrime camera at CFHT (Boulade *et al.*, 2000, 2002).

In the present paper, we describe the software we have developed to reach this goal. Section 2 briefly presents the overall architecture of the software with its major components. Section 3 describes in more detail the moving-object detection portion. In section 4 we present tests performed on two mosaic data sets obtained using the CFH12K camera at CFHT. In particular, we prove that our simultaneous use of two independent approaches greatly reduces the false detection rate, hence reducing the operator workload, while retaining a high discovery efficiency. Finally, in section 5, we give some concluding remarks on the use of this package in a fully-automated processing pipeline.

2 DATA FLOW

A common method to search for TNOs is to acquire several images (typically 3 or more) of a given field of view, separated in time so that the TNOs would move by a few FWHM (Full Width at Half Maximum) between consecutive images. This time interval is of order 1 hour when looking at opposition, since TNO angular motions projected on the sky vary from $\sim$2 to 5 arcsec/hour for TNOs located 30 to 60 AU from the Sun. This time can be extended to 24 hours when looking between 95 and 105 degrees from the Sun where TNOs are almost stationary. In most cases, discovery observations are obtained within 30 degrees from opposition since this increases the retrograde motion of the TNOs, and provides a cleaner separation in apparent motion from main-belt asteroids. For visual searching 2 images of each field can be sufficient (Millis *et al.*, 2002). But for nearly–automated detection near the signal-to-noise limit of the cameras, 3 images is a practical minimum in order to prevent large numbers of false detections.

The first step of data processing consists of removing instrumental signatures (bias and dark subtraction and then flat-fielding) and is not part of the data-reduction pipeline *per se*. Depending on the particular operating procedure of each telescope, one can either do the pre-processing oneself, or obtain already pre-processed images. The pipeline itself starts with 3 pre-processed images of a given field, acquired with an appropriate time interval between each frame.

The general architecture of the detection pipeline is presented in Fig. 1. The individual steps are as follow.

- In each image, detect a few well-behaved (circular) and isolated stars, using the wavelet transform detection code (see section 3). These stars are then used to determine the seeing (FWHM) and a model of the PSF (Point Spread Function). The FWHM will be needed as an input parameter for both detection codes, and an accurate measure of its value improves efficient detection of faint objects. However, for the initial pass which finds the brightest stars an accurate estimate of the FWHM is unimportant, and we use a generic value of 4.5 pixels for the PSF star detection. If the night is photometric, we use the PSF information to plant artificial moving objects in the images in order to characterize our detection efficiency at the same time as we detect new objects. To do this, we choose a random set of magnitudes (usually between 20 and 26 for TNO work), a random set

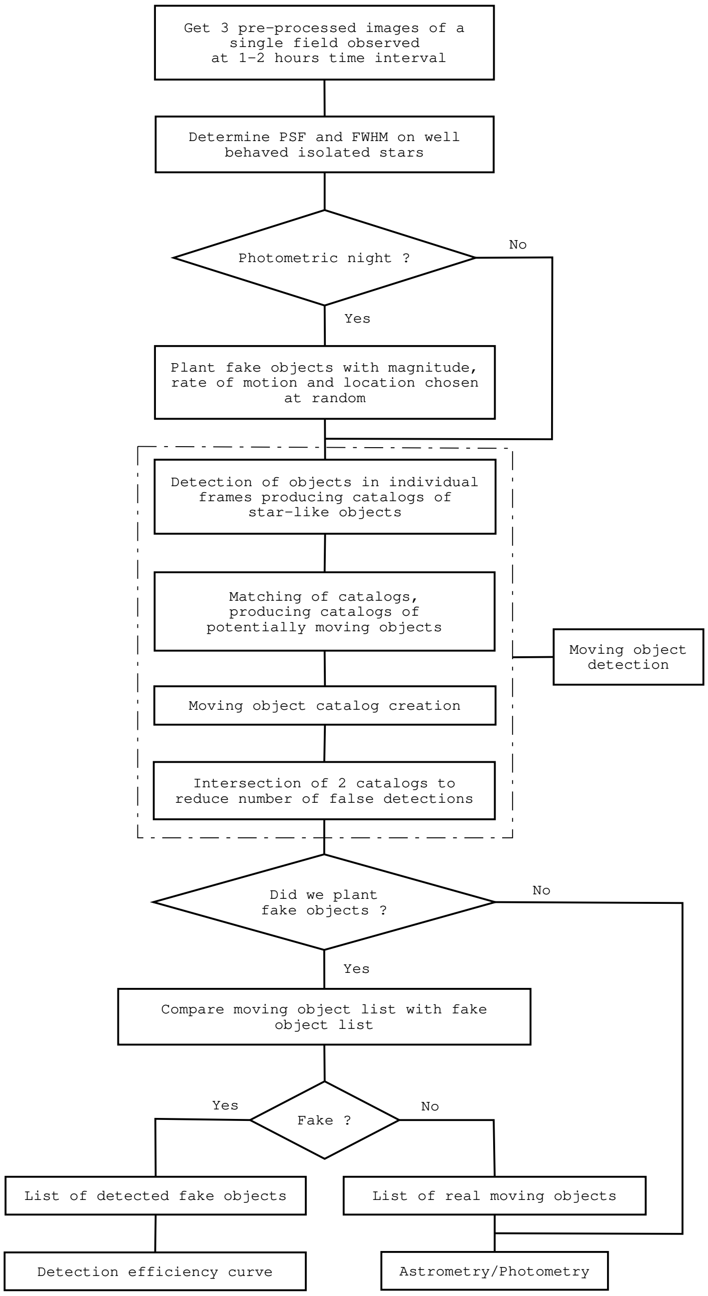

Figure 1. Flowchart of the processing pipeline. Details of the 'moving object detection' steps are explained in section 2.

of rates of motion in the 1"/h to 10"/h range (corresponding to objects between 20 and 150 AU from the Sun), and random orientations within 25-30 degrees from the ecliptic. We have implemented this portion of the package using the publicly-available image reduction package IRAF developed by NOAO.

• The next step is to attempt to detect all moving objects in the 3 images, be they real Solar System objects or implanted fake objects, as discussed in detail in the next section. This step is performed independently using two different codes that run on each triplet of images. They both produce catalogs of stellar like objects, which are then registered, *i.e.* the coordinates are changed so that any fixed star has the same coordinates in all three images. We then search for linearly-moving objects. This produces two sets of candidate moving objects, one from each detection code. We finally combine the two lists, retaining only the intersection. This technique drastically reduces the number of false detections, while keeping roughly the same detection efficiency.

- Up to this point the algorithm has no need for human intervention. At this stage it is necessary to have an "operator" visually confirm all candidates, rejecting those rare false candidates which both codes have incorrectly detected.
- If artificial objects had been implanted, one compares the list of all confirmed candidates with the list of implanted objects, creating a list of detected implanted fake objects. From that list, we can now derive our efficiency as a function of magnitude or rate of motion. It is important to note here that the artificially-implanted TNOs have been subject to an identical procedure. This is important in the presence of false detections, because if artificial objects are implanted in a separate second pass and a human operators does not confirm the candidates from this second pass (but rather objects 'detected' by the codes are just matched to the planted list without operator confirmation) one can overestimate the real-object detection efficiency near the limit.
- Once the detection code has run, one still needs to determine a reasonable estimate of the magnitude scale, *i.e.* an estimate of the photometric zero point, in order to rescale the efficiency curve. Obviously one would like to know the photometric zero points BEFORE implanting the artificial objects; we have often been working in real time at the telescope where we do not know the night's value for the system zeropoint since the standards have not been reduced. However, it is simple enough to use an approximate zeropoint for the system and then correct the magnitude entry of all implanted objects by the magnitude offset between the approximate and final value of the zeropoint.
- For all the real detections, one also needs to determine its astrometric position. We do this by identifying a set of stars from a stellar catalog (recently we have used the USNO-A2.0 catalog) in our images, and then constructing a plate solution (whose order is adapted to the camera system in question) in order to calculate the position of the moving target on the 3 frames. On photometric nights, an aperture correction photometry is also performed.

3 DETECTION OF MOVING OBJECTS

When we first attempted automatic moving object detection, we immediately realized that any automated code would face a problem of efficiency versus false detection rate. A low false-detection rate can be achieved by detecting only high signal-to-noise (S/N) ratio objects, but this results in a very low efficiency in detecting the available moving objects on the frame, especially in cases like the Kuiper Belt where the steep luminosity function results in the majority of the objects being present in the last magnitude above the frame limit. On the other hand, pushing to lower S/N generates a faster than exponential increase in the number of sources on any given frame, resulting in a corresponding dramatic increase in the number of false detections due to fortuitous alignments of noise clumps (see Fig. 2 for an example of how the number of detection increases when the detection threshold decreases). One solution, implemented by Delsanti *et al.* (2000), is to increase the number of images used for detection. Because linearity of motion is used to select the candidates, one needs at least three images,

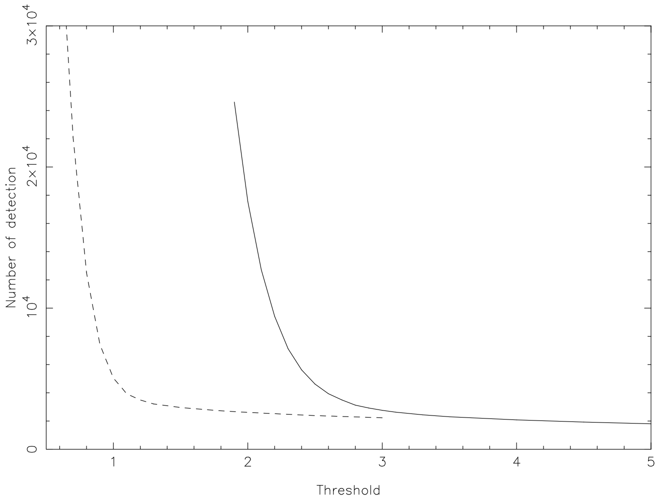

Figure 2. Number of detection in step 1 as a function of the threshold used for S-Extractor path (dashed line) and Wavelet path (solid line)

but imposing linear candidate motion over even more images rapidly decreases the number of false detection as we increase the number of images. One can increase this number at will, at the expense of reducing the sky coverage given a fixed amount of available observing time. In addition, one must balance the advantages of lower false-detection rate versus the loss of objects via confusion with fixed stars and galaxies on one or more of the required exposures.

We decided on a different approach to solve this problem. We began with two different stellar-object detection codes. One is the publicly-available package S-Extractor (Bertin and Arnouts, 1996; see http://terapix.iap.fr/soft/sextractor for details), which detects objects in the normal image space. In contrast, the second package has been developed by the present authors, and detects objects in wavelet space (see below for details). Because the two codes use completely different approach, the false detections they obtain (the noise that they detect as possible stellar-like objects near the detection limit) have very different characteristics. However, both codes detect a similar set of real objects near the limit. So we decided to run the two codes independently, and use their output to produce two lists of candidate moving objects (see subsections 3.2 and 3.3). The part using S-Extractor will be called the *S-Extractor path* in what follows, the part using the wavelet transform the *Wavelet path*. At the end of the S-Extractor path and the Wavelet path, we consider the intersection of the candidate moving-object lists proposed by the two codes. We will see in Sec. 4 how this decreases the number of false detections, while retaining almost the same efficiency. Fig. 3 schematically represents the detection part of the dataflow described in Sec. 2.

3.1 Step 1: star-like object detection

The first step consists of creating a list of star-like objects from each image. For each detected object this list contains the coordinates in the CCD's pixel system and some morphological information, such as an estimate of the flux,

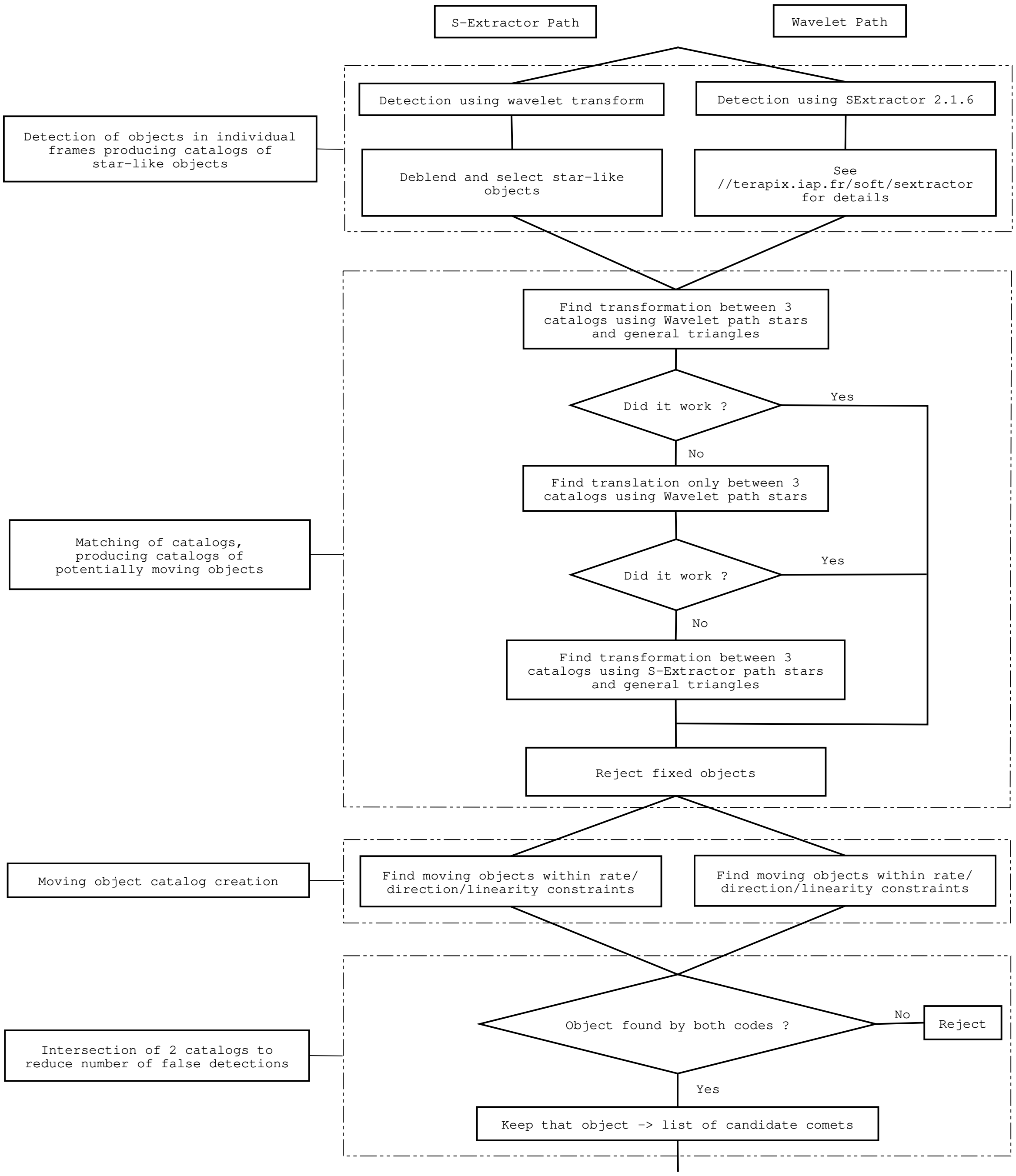

Figure 3. Details of the detection part of the pipeline. Two different detection codes are used and their results are intersected to select the moving object candidates.

the size of the detected object, the maximum intensity, and shape parameters.

3.1.1 S-Extractor path

This step is performed by running S-Extractor with a detection and analysis threshold of 1.3. The detection is done by looking directly at the image pixel values, in real space. This generates a first temporary object catalog. To try to eliminate non-stellar sources and cosmic rays, we reject any source that have an elongation larger than 10 (low-angle of incidence cosmic rays, some galaxies, satellite trails), or that has a ratio of the maximum intensity in a single pixel to the total flux larger than $5/\mathrm{FWHM}^2$. However, all objects with a maximum intensity above background smaller than 100 counts (for CFH12K) are kept as the elongation and/or total flux may be poorly estimated in such cases. Cosmic ray rejection is not critical as the linearity criterion efficiently removes most problems associated with these events. These steps produces the star-like object catalog for the S-Extractor path.

3.1.2 the Wavelet path

The Wavelet path detection step relies on a wavelet decomposition of the images (see Daubechies, 1988 and Mallat, 1989 for a complete description of the wavelet decomposition theory). However, we do not perform a full multiresolution analysis of the signal, nor do we use a reconstruction algorithm before detecting the objects. Rather, we use the wavelet approach to extract the significant signal at the scale corresponding to the FWHM, and then detect the objects in that wavelet space. Such an approach, with Gaussian wavelets, was used for near-Earth asteroid detection in the ODAS survey (Scholl *et al.*, 1999a, 1999b). However, the use of a Gaussian filter resulted in a long computation time, incompatible with our aim of almost real-time analysis.

We therefore switched to a linear wavelet basis, with diadic decomposition (see Bendjoya *et al.*, 1993 and Bury *et al.*, 1996 for 1-D signal analysis using this wavelet decomposition), resulting in the necessary reduction of computing time. In order to calculate the wavelet transform of our signal, we first iteratively apply a set of filters with successively doubled scales. We then compute the difference between two successive smoothed signals to obtain the wavelet transform at a given resolution. More precisely, let $S_0(x_i, y_j)$ be our signal. We define the filtering kernel as:

$$\Phi_0 = 0.5, \qquad \Phi_{-1} = \Phi_1 = 0.25. \tag{1}$$

Given a smoothed signal $S_l(x_i, y_j)$ (S_0 is the unsmoothed, original signal), we compute the next smoothed signal S_{l+1} by applying the filtering kernel:

$$\tilde{S}_l(x_i, y_j) = \sum_{r=-1}^{+1} \Phi_r S_l(x_{i+r*k}, y_j), \tag{2}$$

$$S_{l+1}(x_i, y_j) = \sum_{r=-1}^{+1} \Phi_r \tilde{S}_l(x_i, y_{j+r*k}), \tag{3}$$

where $k = 2^l$. The wavelet transform, or more precisely the wavelet coefficients at scales from 2^l to $(2^{l+1} - 1)$ is obtained by taking the difference between S_{l+1} and S_l:

$$C_l(x_i, y_j) = S_{l+1}(x_i, y_j) - S_l(x_i, y_j). \tag{4}$$

In theory, one could compute the wavelet coefficients for scales up to the size of the data set. In practice, we are interested in the scale corresponding to the FWHM, hence we compute the $C_l(x_i, y_j)$ for scale:

$$l = E\left(\frac{\ln(FWHM)}{\ln(2)}\right), \tag{5}$$

where $E(x)$ represents the integral part of x, i.e. the largest integer not greater than x. We iteratively compute the smoothed signals S_l and S_{l+1} to produce the wavelet coefficients C_l, permitting the computation of the mean $\langle C_l \rangle$ and the standard deviation σ_{C_l} of the wavelet coefficients. One then selects all pixels with a wavelet coefficient larger than $\alpha\sigma_{C_l} + \langle C_l \rangle$, where α is a user given threshold, usually $\simeq 2.7$ for CFH12K. We segment the wavelet coefficients into sets of contiguous selected pixels, which represent the detected objects. For each detected object, one calculates all momenta up to second order for the actual signal, that is, the image pixel value minus the local background. Finally, we reject all objects that are either too small (less than 4 pixels), too elongated (aspect ratio larger than 10), or look like cosmic rays (using the same criteria as previously discussed). The remaining objects constitute the catalog of star-like objects for the Wavelet path.

3.2 Step 2: selection of potentially moving objects

At this point each path has produced a catalog of objects which lists for each object its image coordinates (x, y), the number of selected pixels, the total flux above background for all selected pixels, the maximum pixel intensity, and the aspect ratio or elongation of the object. The aspect ratio or elongation of the object is the ratio of the small axis to the long axis of the ellipse defined from the second order momenta of the signal.

At this point each catalog has a different coordinate system because the telescope does not return to precisely the same position for the three exposures. We must therefore register the catalogs, *i.e.* find the function that will change one coordinate system into that of a reference image. For each set of three images, we consider the first image as the reference image, and seek the two transformations to match the second and third images to the first one.

To this end, for each pair of images, we select a few tens of bright stars. We take the brightest ones in the catalog with a maximum intensity less than 80% of the largest intensity ever recorded for that image, in order to avoid stars near the CCD's saturation limit. From this set, we keep only those that are at least 40 pixels away from any other brighter star, and and least 2 FWHM away from any other detected object. Departing from the philosophy utilized up to now, we use the same transformations to register the catalogs for both paths, with the rationale that there is only one such transformation for each pair of images. To insure that we correctly find it, we apply several algorithms sequentially until one of them finds the transformation, instead of using them in parallel.

We first use our own implementation of the algorithm described by Valdes *et al.* (1995). This algorithm uses the bright stars selected from the the Wavelet path catalogs to creates a list of triangles for each image. Searching for similarities in the shape of the triangles, it obtains a set of corresponding stars. Using these corresponding stars, it finally uses a least-squares fit to find the best transformation made of translations, rotations, and magnifications. After applying the transformation to the selected stars of the second image, we compile the statistics of their distances to the corresponding star in the first image, rejecting stars for which the distance is larger than 3 standard deviations and then iterate the procedure. The algorithm comes to an end when we stop rejecting stars, in which case the process converged and we have the desired transform, or there is no longer enough stars (less than 2), in which case the algorithm failed.

If the first method failed, we attempt a less general, slower, yet more robust algorithm, still using the the Wavelet path bright stars. Here we assume that the orientation and scale of the frame is the same for all images and hence search for solely a translation between images. We construct triangles from bright stars from each image. For each triangle in the reference image, we search for a triangle in the other image whose vertices are obtained by the same translation of those from the reference triangle, to within a certain tolerance, 0.3 pixel for CFH12K images. This gives us a set of possible translations. We compute the mean and standard deviation of them. We reject those that are further than 10 standard deviations from the mean deviations, repeating this process until no more translations are rejected. Here again, the algorithm fails if there are less than two triangles left.

If this second method also fails, we go back to the triangle matching approach, but this time we use the publicly available package `match`, written by Michael Richmond, based on the Valdes *et al.* (1995) algorithm. In this case, we apply it to the bright stars from the the S-Extractor path catalogs.

In our experience, this three-algorithm attack never failed to produce the correct transform. Applying the transformation just derived, we obtain three catalogs of registered stars and objects based on the pixel coordinate system of the first image. We then identify all objects that appear at the same position (to within a tolerance of one FWHM) in all three images. If they are similar in flux and shape, they are tagged as fixed (unmoving) stars or galaxies. We remove the fixed objects from the catalogs, obtaining three catalogs in which to search for moving objects.

3.3 Step 3: finding moving objects

For objects moving fast enough to be detected in one night of observation, the apparent motion with respect to the background of fixed stars over the length of the night is almost linear with respect to time. Selecting real moving objects from all possible triplets of candidates is done by verifying that the positions in the candidate triplet vary linearly with time. This is done by selecting an object in the first image, one in the third image, interpolating the position the object should have on the second image, and comparing to the actual position. If the distance is less than some value, then the motion is declared linear. Candidate triplets are restricted to the range of possible motions of real objects in the Edgeworth-Kuiper belt, or the Centaur region, both in speed and in direction. Typical speeds for an object at 100 AU from Earth, observed at opposition, is $\simeq$1.5"/hr, while an object at the distance of Uranus has a speed of $\simeq$8"/hr. When observed at opposition most of the apparent motion of the objects is due to the motion of Earth, hence the direction of the motion is within a few degrees of the ecliptic plane. In practice, we only consider triplets with a speed in the range 1"/hr to 10"/hr, with a direction within 30 degrees from the ecliptic direction. This range is larger than what is strictly needed to accommodate for uncertainties in the position on short displacements. When a triplet is found to have a linear motion within the given range, we check that all three objects have a similar flux and shape. If so, the triplet in stored in a catalog of candidate moving objects.

A word of caution is in order here. This procedure is not optimal for objects close to the stationary point, as this requires observations over multiple nights and the linearity of motion is reduced. It may even go from retrograde to prograde or vice-versa over the course of the observations. Relaxing the linearity check yields many more false candidates, which poses practical difficulties. As stated earlier, this code will perform best when used to search for small bodies in data acquired within 20 degrees of opposition.

Two independent implementations of the previous algorithm have been done, one for the S-Extractor path and one for the Wavelet path. Each implementation uses the same range of angular velocities, but the actual tests for linear motion and similarity of flux and shape are tuned to the exact quantities provided by each detection code. The nonlinearity tolerance on the position in the second image is one FWHM for the Wavelet path, and 3 pixels for the S-Extractor path. The direction of motion is checked directly on the displacement from the first to the third image for the S-Extractor path. For the Wavelet path, the displacements from the first to the second image and from the second to the third image are both independently checked against the allowed range of directions.

Once each path has produced a catalog of candidate moving objects, we compare them, and retain only objects appearing in both catalogs. Two objects are considered identical if the positions from the S-Extractor path and the Wavelet path are within one FWHM from each other at all three times. This yields a combined catalog of candidate moving objects. At this point, if artificial objects were added to the images at the beginning of the process, the combined list is compared to the list of implanted objects in the same way as before (1 FWHM tolerance on each image). All artificial objects are thus removed from the combined list, and stored in a catalog of detected implanted objects for determination of the efficiency function. The remaining combined catalog of candidate moving objects is the end product of the detection part of the dataflow.

At this point, the operator will interact with the process, and check for the validity of the detections. The ramifications of doing this interactive work *after* the removal of the artificially implanted objects will be discussed below. For those objects which the human operator accepts as real (rather than false detections) astrometry and potentially photometry are performed.

4 TWO TEST CASES

In July 1999, we had 4 nights of observations at CFHT to search for irregular satellites around Uranus and Neptune (Gladman *et al.*, 2000). The pipeline presented in the present paper is a development of the codes used in Gladman *et al.* (2000) to discover three uranian irregular satellites. The first test we present here uses 4 fields obtained near Neptune on July 20, 1999 UT; in all we present an analysis of 39 CCDs. The exposure time was 8 minutes, and the seeing was between 4 and 5 pixels, i.e. between 0.8" and 1".

Table 1. Comparison of the number of candidate moving objects (column 2), detected artificial objects (column 3), proposed real candidates (column 4), detected real moving objects (column 5) and false detections (column 6). The first line refers to the S-Extractor path in the detection algorithm, the second line to the Wavelet path, and the third line to the combination of both paths.

Path	Moving cand.	Art. objects	Cand.	Real obj.	False det.
S-Extractor	3 162	921	2 241	4	2 237
Wavelet	1 532	937	595	3	592
Combined	861	841	20	3	17

In Fig. 4 we present the efficiency curves for the S-Extractor path and the Wavelet path separately, and then for the intersection of both codes. These curves show the generic behavior that there is a near plateau for bright magnitudes with a rapid drop off starting about 1 magnitude above the very limit of detection. The fact that the efficiency never reaches 100% is caused by the fact that background stars and galaxies cover roughly 10–20% of the sky at these magnitudes; note that pencil-beam methods (Gladman *et al.*, 2001) do not suffer from this effect because there is sufficient data to remove the signal from the stationary objects. For efficiency curves of this style it is customary to define a '50% efficiency' not at the level where the absolute efficiency is 50% but rather where the efficiency drops to half of it's maximum value (otherwise a survey in a crowded stellar field could have a maximum efficiency of *less than* 50% and thus no 50% level!). In this test the 50% efficiency level is reached at $m_R = 24.0$ for the S-Extractor path, at $m_R = 23.85$ for the Wavelet path, and at $m_R = 23.8$ for the combination.

Thus although there is an efficiency decrease during the combination, the effect is mild. Much more important, as Table 1 shows, taking the intersection has made the number of false detections, and hence the operator work load, a factor of 30 to 200 smaller. To reject the false candidates proposed by the codes, the operator has to blink the 3 images in a small region around each candidate. Assuming that such a check can be achieved in an average of about 30 seconds (including time to switch to the right location, display images, blink them, and loss of efficiency when tired), the operator will spend about 10 minutes verifying the outcome of the combined codes for the 39 CCDs searched. If we wanted to check the outcome from each code separately, this would require almost 19 hours for the S-Extractor path, and 5 hours for the Wavelet path. For the purposes of this test we did indeed perform these visual tests and in our experience the time per candidate increases with the number of candidates as the operator get tired; in addition, due to the tedious nature of the work operators can often make errors which declare correct candidates (real or artificial) to be false detections and vice versa.

It can be seen from Table 1 that the S-Extractor path identified one more real object than the Wavelet path (and hence the combined code); this object had an R-band magnitude of about 24, where the efficiency is about 0.3. This discrepancy is well within the Poissonian error bars, and it does not change significantly the estimated total population. And although here we performed visual check of all candidates for the sake of verifying the efficiency of our codes, this would be impracticable in a real survey.

Table 2. Same as Table 1, but for data taken in November 2002, with the upgraded CFH12K camera.

Path	Moving cand.	Art. objects	Cand.	Real obj.	False det.
S-Extractor	11 188	10 577	611	5	606
Wavelet	12 520	11 104	1 416	5	1 411
Combined	9 858	9 827	31	5	26

At the end of November 2002, we obtained 3 observing nights at CFHT, with the upgraded CFH12K camera. The second test we present uses 7 different fields acquired on November 30, 2002 UT, close to ecliptic, and near opposition, consisting of triplets of images of 84 CCDs. The exposure time was 90 seconds, and the seeing was between 2.5 and 3.5 pixels, i.e. between 0.5" and 0.7". Fig. 5 is similar to Fig. 4, but for this new data set. Here again, the efficiency of the combined code is only slightly impaired when compared to each path independently. But, as can be seen from Table 2, this small decrease in efficiency again dramatically decreases the number of false candidates. The actual number of false detection for each path separately, and for the combined code, will depend on the conditions of observations and to some small extent the parameters used in the code, but the typical number of false detection is well represented by these two examples. There is a gain of a factor of up to several hundreds when compared to the union of the results from both paths (under the hypothesis that all potential candidates from both codes are inspected).

When comparing Fig. 4 and 5, note that the error bars are much smaller in the second one because the number of fake objects added to the images was 1677 in the first test, and 16800 for the second. One notes also that the 50% magnitude (defined above) is about the same for both cases. It turns out that the better seeing compensated for the decrease in exposure time.

5 DISCUSSION

In this paper, we have described in detail the core part of our automated Moving Object detection Pipeline (MOP) which will be used for data processing of the Ultra-Wide component of the CFHT Legacy Survey (ecliptic component). The detection portion of the package (without artificial object implantation nor the astrometric portions) is freely available from the World Wide Web server of Besançon Observatory (http://www.obs-besancon.fr/publi/petit/Preprints/detection.html), without guarantee or support. It can be used as a stand-alone pack-

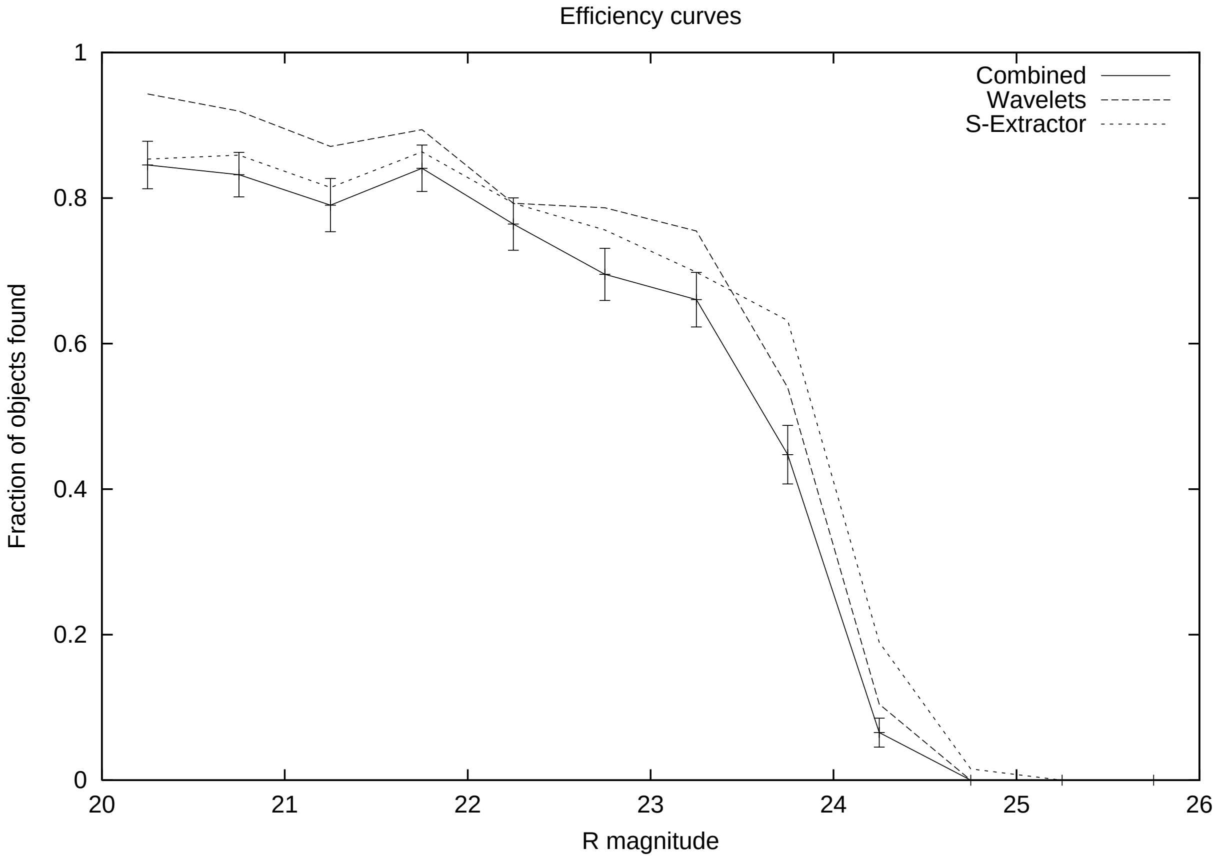

Figure 4. Comparison of the efficiency curves for images acquired in July 1999, with the old CFH12K camera during a Neptune irregular satellite search. The fraction of objects identified is shown as a function of moving-object magnitude. The solid line show the efficiency curve for the combined detection code, the error bars corresponding to 1 sigma, assuming a Poissonian distribution, which is dominated by the finite number of objects implanted in the data. The short-dotted line is for the S-Extractor path, and the long-dashed line for the Wavelet path.

age to semi-automatically search triplets of images for moving objects. We have shown in section 3.3 how our approach can drastically reduced the amount of operator work, while retaining nearly the same detection efficiency.

We have compared the efficiency, execution time, and false-detection rate of our detection package to the only other publicly-available package known to us, designed by Rousselot *et al.* (1999), which runs inside the public-domain MIDAS package developed by ESO. Running our complete detection code on a 1 GHz Athlon PC, one needs a little less than 45 minutes for the 39 triplets of the first test, or 70 seconds per triplet, not counting operator inspection of course. We ran the Rousselot *et al.* code on a single triplet, with different values of the detection threshold. On that particular triplet, our code found 15 objects, all artificially-implanted, and had no false detections; remember however that one can expect one false detection per chip on average. Results for the Rousselot *et al.* code are given in Table 3. If one want to have an efficiency close to ours, the Rousselot *et al.* code takes 6 times longer to run, and the extra operator time to verify the validity of the candidates is a few tens times larger.

The alert reader may have been troubled by one step of the above procedure. In order for the survey to be precisely

[htbp]

Table 3. Timing and efficiency of Rousselot *et al.* code on a single triplet, compared to our moving-object pipeline (MOP). The first column gives the Rousselot *et al.* threshold for detection in ADU (except for the last line corresponding to our code), the second column gives the time for running the detection code on a 1 GHz PC, the third column the number of objects detected, and the fourth column the number of false detections.

Threshold (count)	time (sec)	Objects found	False detections
Rouss-80	170	11	32
Rouss-50	425	14	60
MOP	70	15	0

characterized in terms of magnitude efficiency one must pass the artificial and real objects through *exactly* the same filter. However, we have removed the artificial objects from

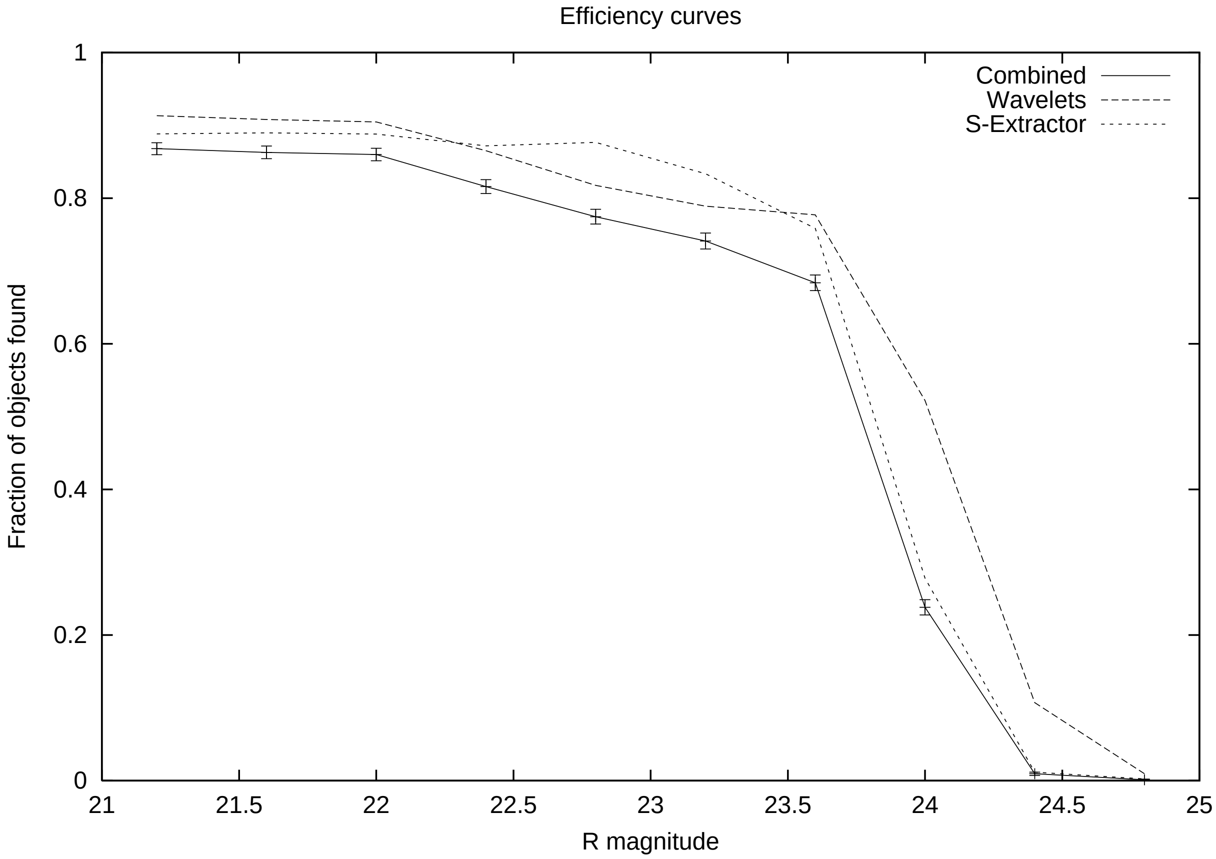

Figure 5. Same as Fig. 4, but for data taken in November 2002, with the upgraded CFH12K camera.

the process after the automated pipeline has run but *before* the operator is shown the candidates (thus consisting of false detections and real moving objects). This could be a source of concern because, in an attempt to reach as deep into the noise as possible, one pushes the detection threshold of an automated code down to where a steep increase in the number of detections occur (as in Fig. 2), with a lower limit essentially determined by how many false detections the 'operator can handle'. However, we immediately observed that one *can* actually find fainter and fainter planted objects by pushing the threshold down, albeit at the price of a 'super-exponential' increase in the number of false candidates; in fact, one finds objects that the operator cannot reliably say are real. The result of this is that *if* one pushes down the threshold to the point where large numbers of false detections are being generated then one MUST have the operator blink through the entire candidate set (including the artificial objects); if not, the following insidious effect occurs: In order to have an accurate determination of the efficiency one plants large numbers (hundreds) of artificial objects in each CCD, and in order to reduce the operator workload one wants to *not* have the operator blink and reject these false candidates. However (Fig. 6), near the limit the operator will reject artificial objects at low signal-to-noise as unreal, meaning that the operator-determined (and thus correct, as it applies to the real moving objects) efficiency is *lower* than one in which the all objects blindly located by the code are counted as detections. Although this effect may not seem dramatic (Fig. 6), Fig. 7 shows that near the limit this can result in a mis-determination of the efficiency function by a factor of 2, and is thus especially important for determining the luminosity functions of steep size distributions when most of the detected objects are in the faintest bins.

Various versions of this algorithm have been used since summer 1999 on images obtained on different telescopes, and it allowed us to discover or recover already hundreds of TNOs and irregular satellites.

ACKNOWLEDGMENTS

We thank Philippe Rousselot for his help in using his software package. Lynne Allen has served as a helpful tester of some portions of the pipeline software. This work was supported by grants from the ACI and PICS program of the French government. M. Holman acknowledges support from NASA Origins grant xxxxx-xxx ????

REFERENCES

Allen, R. L., Bernstein, G. M. & Malhotra, R. 2002, Astron. J., 124, 2949
Bendjoya, Ph., Petit, J-M. & Spahn, F. 1993, Icarus, 105, 385
Bertin, E. & Arnouts, S. 1996, Astron. Astrophys., 117, 393
Boulade, O., Charlot, X., Abbon, P., Aune, S., Borgeaud, P., Carton, P.H., Carty, M., Desforge, D., Eppellé, D., Gallais, P.,

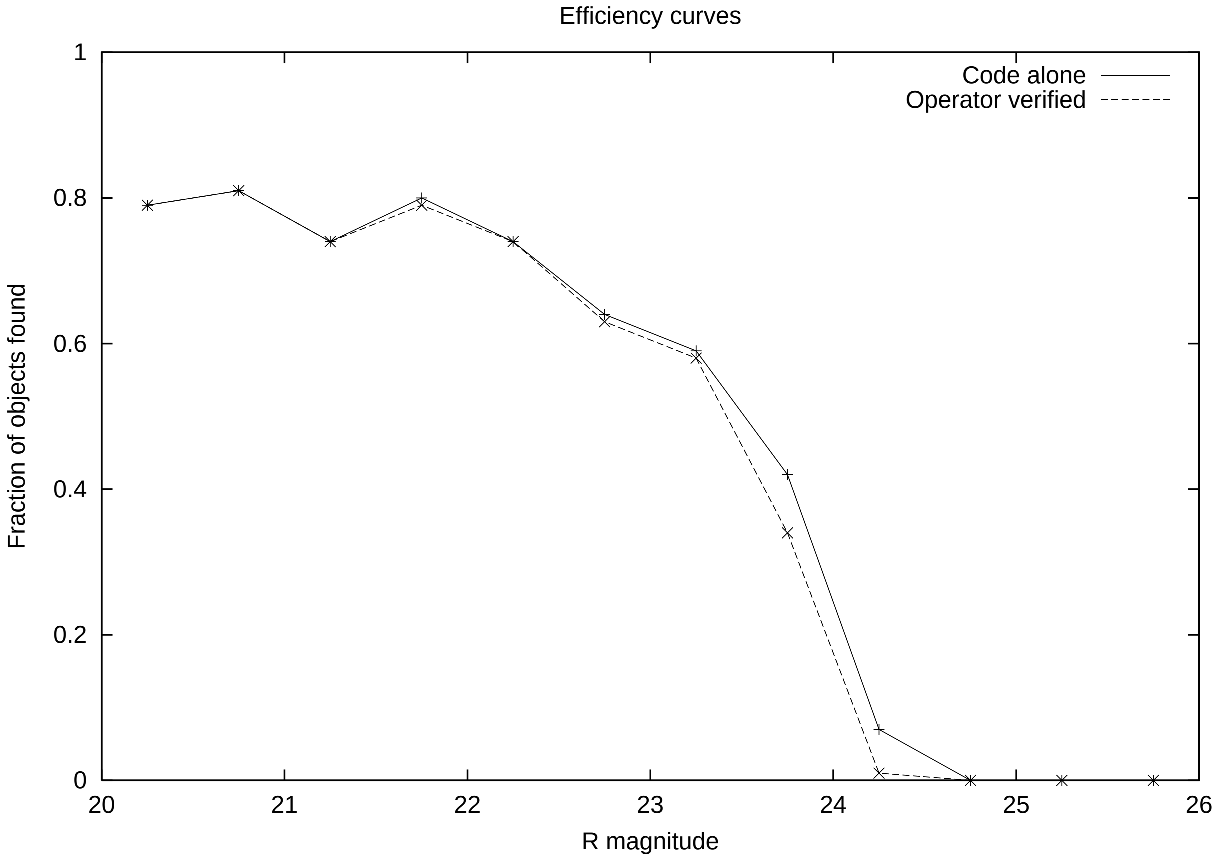

Figure 6. An illustration of the effect of removing the artificial objects before and after operator verification. The solid line shows the fraction of objects found when the code runs 'blind' and the fraction of objects found is built by calculating what of the artificial objects in the planted list is found as a function of magnitude without a human operator involved. The dashed curve shows the efficiency if calculated as the fraction of detected objects that have passed operator verification. See text for discussion.

Gosset, L., Granelli, R., Gors, M., de Kat, J., Loiseau, D., Mellier, Y., Ritou, J.L., Roussé, J.Y., Starzynski, P., Vignal, N. & Vigroux, L. 2000, In *Proc. SPIE 4008*

Boulade, O. & the MegaCam Team 2002, In *DSM-DAPNIA internal reports* http://www-dapnia.cea.fr/Phys/Sap/Activities-/Projets/Megacam/Presentations/megacam_ccdreport.ps.gz

Bury, P., Ennode, N., Petit, J-M., Bendjoya, Ph., Martinez, J-P., Pinna, H., Jaud, J. & Balladore, J-L., 1996, Nuclear Instruments and Methods in Physics Research, Section A, 383, 572

Daubechies, I. 1988, Comm. Pure Appl. Math., 41, 909

Davis, D.R. & Farinella, P. 1997, Icarus, 125, 50

Delsanti, A., Hainaut, O.R., Boehnhardt, H. & Delahodde, C.E. 2000, American Astronomical Society, DPS meeting

Duncan, M.J. & H.F. Levison 1997, Science, 276, 1670

Farinella, P., Davis, D.R. & Stern, S.A. 2000, In *Protostars and Planets IV* (V. Mannings, A.P. Boss and S.S. Russell, Eds) p 1255. Univ. of Arizona Press, Tucson

Gladman, B., J.J. Kavelaars, P. Nicholson, T. Loredo & J.A Burns 1998, Astron. J, 116, 2042

Gladman, B., Kavelaars, JJ., Holman, H., Petit, J-M., Scholl, H., Nicholson, P. & Burns, J.A. 2000, Icarus, 147, 320

Gladman, B., Kavelaars, JJ., Petit, J-M., Morbidelli, A., Holman, M.J. & Loredo, T. 2001, Astron. J., 122, 1051

Irwin, M., Tremaine, S. & Zytkow, A.N. 1995, Astron. J., 110, 3082

Jewitt, D.C., J.X. Luu & C. Trujillo 1998, Astron. J., 115, 2125

Kenyon, S. 2002, PASP, 114, 265

Larsen, J.A., Gleason, A.E., Danzl, N.M., Descour, A.S., McMillan, R.S., Gehrels, T., Jedicke, R., Montani, J.L. & Scotti, J.V. 2001, Astron. J., 121, 562

Levison, H.F. & Duncan, M. 1990, Astron. J., 100, 1669L

Mallat, S.G. 1989, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, 674

Millis, R.L., Buie, M.W., Wasserman, L.H., Elliot, J.L., Kern, S.D. & Wagner, R.M. 2002, Astron. J., 123, 2083

Petit, J-M., A. Morbidelli, A. & Valsecchi, G. 1999, Icarus, 141, 367

Rousselot, P., Lombard, F. & Moreels, G. 1999, A&A, 348, 1035

Scholl, H., Bijaoui, A. & Savalle, R. 1999a, In *Treasuring Hunting in Astronomical Plates* (P. Kroll, C. la Dous and H.-J. Braeuer, Eds), pp 188-190. Acta Historica Astronomiae 6

Scholl, H., Bijaoui, A., Hahn, G., Hoffmann, M., Maury, A. & Mottola, S. 1999b, In *Atelier GAIA* (M. Froeschlé and F. Mignard, Eds), pp 29-34. Observatoire de la Cote d'Azur

Stern, S.A. & Colwell, J.E. 1997, Astrophys. J., 490, 879

Trujillo, C. & Jewitt, D. 1998, Astron. J., 115, 1680

Trujillo, C.A. & Brown, M.E. 2001, Astrophys. J., 554, L95

Trujillo, C.A., Jewitt, D.C. & Luu, J.X. 2001a, Astron. J., 122, 457

Trujillo, C.A., Luu, J.X., Bosh, A.S., & Elliot, J.L. 2001b, Astron. J., 122, 2740

Valdes, F.G., Campusano, L.E., Velasquez, J.D. & Stetson, P.B. 1995, PASP, 107, 1119

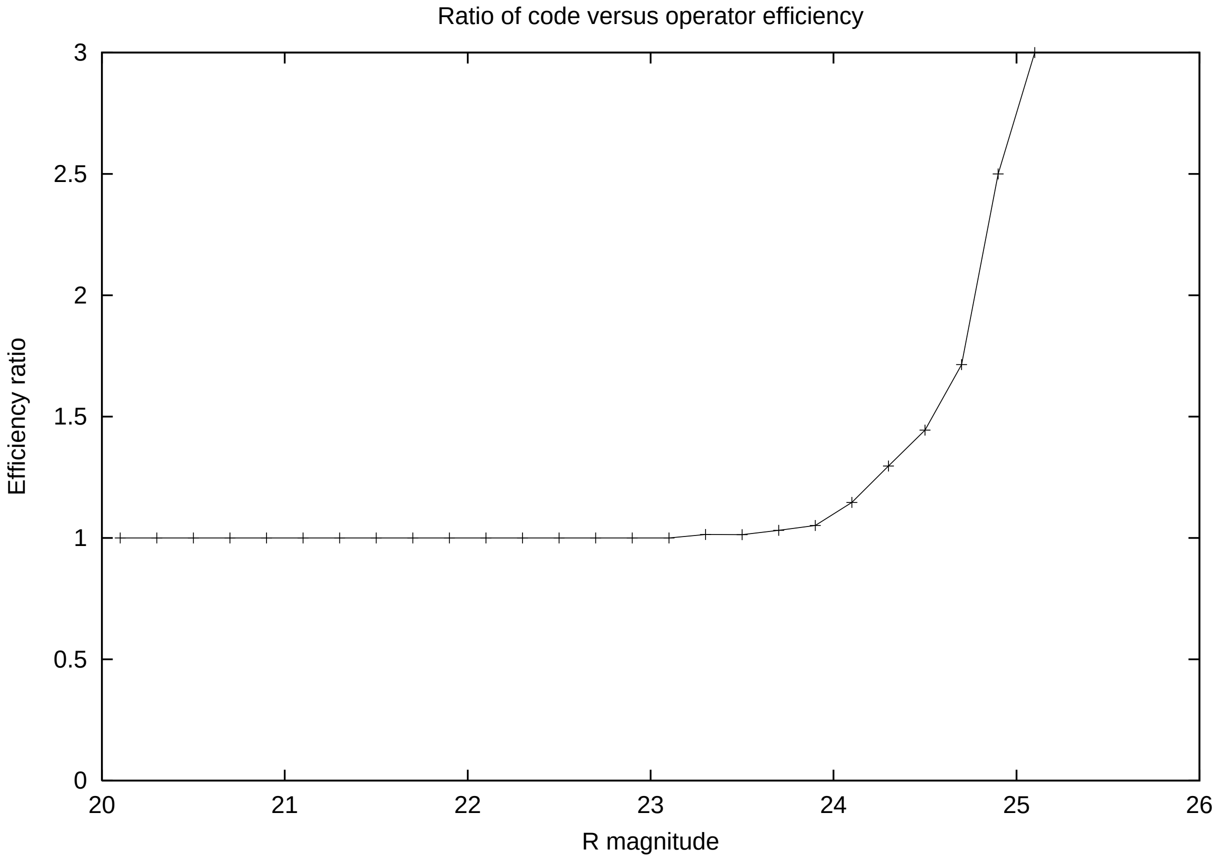

Figure 7. The difference in efficiency determination when the operator does not verify the code's detection of the artificially-implanted objects. The plot shows the ratio of Fig. 6's "Code alone" versus the "Operator verified" entries. Near the detection limit the uncertainty in the efficiency function exceeds a factor of two.