# Moving Target Detection Using Image Correlation Postprint

**Authors:** Pan Zuheng, Peng Qingyu, Lu Xing, Chen Xiao, Li Dan

**Date:** 2022-10-21T11:43:55+00:00

## Abstract

In the field of astronomical image processing, image subtraction techniques have found extensive applications. Due to factors such as different atmospheric conditions, exposure times, and filters, direct image subtraction between two images is not feasible. This paper proposes a novel image subtraction algorithm, which fundamentally utilizes correlation in a statistical sense to eliminate portions with similar flux distributions between two images while preserving portions with dissimilar flux distributions. The algorithm is characterized by very rapid execution, numerical stability, and local independence. Based on the algorithm presented herein, an image subtraction code was developed using Python as the interface language and C as the underlying implementation. This algorithm, along with three other similar algorithms, was employed to process astronomical images. Experimental results demonstrate that the algorithm can identify differences between two images within a very short timeframe and rapidly locate moving targets, while exhibiting favorable robustness and stability in positional measurement.

## Full Text

## Searching for Moving Objects Using Image Correlation

**Pan Zuheng[12], Peng Qingyu[12], Lu Xing[123], Chen Xiao[12], Li Dan[12]**

[1]Department of Computer Science, Jinan University, Guangzhou, Guangdong 510632, China
[2]Sino-French Joint Laboratory for Astrometry, Dynamics and Space Science, Jinan University, Guangzhou, Guangdong 510632, China
[3]Department of Physics, Jinan University, Guangzhou, Guangdong 510632, China

**Abstract:** Image subtraction techniques are widely employed in astronomical image processing. However, due to varying atmospheric conditions, different

exposure times, and different filters, two images cannot be directly subtracted from one another. This paper proposes a novel image subtraction algorithm that fundamentally uses correlation to eliminate, in a statistical sense, portions with similar flux distributions between two images while preserving portions with dissimilar flux distributions. The algorithm executes extremely rapidly, is numerically stable, and operates locally. Based on this algorithm, we have developed an image subtraction code with Python as the interface and C as the underlying implementation. We applied this algorithm alongside three other similar algorithms to process astronomical images. Experiments demonstrate that our algorithm can identify differences between two images and rapidly locate moving objects within a very short time frame, while exhibiting good robustness and stable position measurement.

**Keywords:** correlation; image processing; image subtraction; astrometry; moving objects

## 0 Introduction

Image subtraction plays a crucial role in detecting unknown sources, discovering supernovae, and studying gravitational microlensing phenomena. Tomaney and Crotts first performed image subtraction by solving for convolution kernels in the frequency domain, applying this technique to monitor microlensing events and process data from the M31 galaxy. Kochanski et al. subsequently proposed an optimal image subtraction method based on non-linear least-squares fitting, which entails substantial computational time even for sparse fields when fitting only a small subset of pixels. Alard and Lupton also introduced an optimal image subtraction algorithm that solves for convolution kernels directly in the image spatial domain using standard linear least-squares. This approach decomposes the kernel into multiple Gaussian functions multiplied by polynomials. Alard later refined this method using spatially varying kernels and optimized its computational strategy. Wozniak applied Alard' s algorithm to process 380 GB of OGLE-II bulge microlensing data.

Among these methods, the Alard and Lupton approach remains the most classical. However, it requires users to specify the number of Gaussian basis functions, their associated sigma values, and the polynomial order—parameter choices that inevitably confuse users and demand extensive experimentation to optimize for specific datasets. Building upon Alard and Lupton' s ideas, researchers subsequently developed several similar image subtraction techniques. For instance, Bramich treated the convolution kernel as a discrete pixel array and directly solved for kernel pixel values using linear least-squares, making kernel size the sole adjustable parameter. Yuan and Akerlof employed cross-convolution to avoid requiring high-quality reference images. Both methods, known as deconvolution approaches, evolved from the Alard and Lupton framework but share the common drawback of requiring parameter tuning.

The computational speed of the Bramich method heavily depends on kernel size, becoming prohibitively slow as the kernel grows larger. Moreover, these algorithms are not locally independent, making it difficult to process large images as whole units and typically necessitating division into smaller sub-images. They also suffer from numerical instability, as bad or saturated pixels in the input or reference images can contaminate entire pixel groups within the kernel region. Consequently, these methods usually require removal of bad and saturated pixels before subtraction and should avoid excessively large kernels. These operations increase the complexity of image subtraction and may cause some meaningful pixels to be overlooked. Zackay et al. later proposed an algorithm based on fundamental statistical principles that overturned traditional deconvolution-based image subtraction, offering local independence and numerical stability.

This paper also proposes a less parameter-dependent alternative method based on fundamental statistical principles, performing image subtraction according to flux distribution patterns between input and reference images. Specifically, it utilizes correlation between flux distributions in two small regions at the same location in both images to eliminate similar portions while preserving dissimilar ones. Based on this algorithm, we have developed an image subtraction code with Python as the interface and C as the underlying implementation. This algorithm can search for moving objects within extremely short timeframes, overcoming the difficulties of manual visual inspection. We refer to this approach as "Eliminate Similarity and Preserve Difference," abbreviated as esapd.

## 1 Method

To search for moving celestial bodies, this paper proposes a correlation-based image subtraction technique. Its core principle involves using the Pearson correlation coefficient to measure the correlation strength between two corresponding small regions in the two images. When correlation is strong, differences between the images are reduced; when correlation is weak, differences are preserved.

As shown in Figure 1 [Figure 1: see original paper], subfigure (c) represents the difference image between subfigures (a) and (b), obtained by aligning the two images, subtracting their respective backgrounds, normalizing them (normalization is described below), and then performing direct subtraction. Subfigure (d) shows the correlation map between subfigures (a) and (b) (the method for obtaining the correlation map is described below). Experimental observations reveal that in common background regions, correlation coefficients primarily range from -0.2 to 0.2 (the gray areas in the correlation map), while in common non-background regions, coefficients mainly range from 0.8 to 1.0 (the white areas). Regions containing moving objects (in non-overlapping cases where the moving object does not coincide with other celestial bodies) exhibit weak correlation.

Based on this discovery, we can reduce pixel intensity differences in strongly correlated regions while preserving differences in weakly correlated regions, thereby

highlighting moving objects. The moving object positions are indicated by red circles in subfigure (c), with white objects representing the moving target in subfigure (a) and black objects representing it in subfigure (b).

To obtain the correlation map, we use the Pearson correlation coefficient, calculated as follows:

$$P_{cc}(r,i) = \frac{cov(r,i)}{\sigma_r \sigma_i} = \frac{E[(r-E(r))(i-E(i))]}{\sqrt{E[(r-E(r))^2]E[(i-E(i))^2]}}$$

where $r$ and $i$ represent pixel intensity values in small regions at the same location in the reference and input images, respectively. These regions, called p-kernels, are square in shape. $E(r)$ and $E(i)$ denote the mean pixel intensities within each region, while $\sigma_r$ and $\sigma_i$ represent their standard deviations. $cov(r,i)$ calculates covariance, and $E$ denotes mathematical expectation. $n$ represents the number of pixels in the p-kernel. We use this formula to compute the Pearson correlation coefficient, which ranges from -1 to 1, for the central pixel position of these small regions. For boundary cases, we apply truncation, ignoring pixels outside the boundaries and reducing the number of pixels $n$ in the p-kernel. This yields a Pearson correlation coefficient matrix of the same dimensions as the original image, referred to as the correlation map. Generally, Pearson correlation coefficients less than 0 indicate negative correlation, while those greater than 0 indicate positive correlation. Larger absolute values signify stronger correlation, and smaller values indicate weaker correlation.

To eliminate similar portions between the two images as much as possible while preserving dissimilar portions, we modify the Pearson correlation coefficient values using a modified sigmoid function:

$$S(x) = \frac{1}{1+e^{-x}}$$

$$P' = \frac{1}{1+e^{-1500(P-0.5)}}$$

Equation (3) shows the standard sigmoid function, while equation (4) presents the modified version. Its function graph is illustrated by the red solid line in Figure 2 [Figure 2: see original paper], where the horizontal axis represents original Pearson correlation coefficient values and the vertical axis represents modified values. The black dashed line indicates the identity transformation without modification. The correlation map after applying this modification is shown in subfigure (e) of Figure 1, where the contrast becomes more pronounced, facilitating elimination of similar flux distributions while preserving dissimilar ones. Note that while original Pearson correlation coefficients may be negative, their absolute values are typically small because strongly negative correlations rarely occur between aligned, background-subtracted, and normalized images.

This modified sigmoid function effectively eliminates the impact of negative values.

When moving objects overlap with other celestial bodies, strong correlation may also occur, which is undesirable for preserving differences. To address this issue, we introduce the differential flux coefficient, which operates as follows:

$$diff_{cc}(r,i) = \frac{\sum |r-i|}{\min(\sum r, \sum i)}$$

where $r$ and $i$ represent pixel intensity values in small regions at the same location in the reference and input images, respectively. These regions, called d-kernels, are also square. Boundary handling follows the same principle as for Pearson correlation coefficient calculation. This formula yields the percentage flux change between the two small regions. For example, when the sum of pixel intensities in one d-kernel is 13 and the other is 10, the percentage flux change is 30%. In equation (6), we set a threshold $T$ (detailed in section 3.2). When $diff_{cc}(r,i) \geq T$, $F_{cc}(r,i) = 0$; otherwise, $F_{cc}(r,i) = 1$. We compute the differential flux coefficient $F_{cc}(r,i)$ for the central pixel position using pixel intensities from these small regions, resulting in a differential flux coefficient matrix of the same size as the original image, called the threshold map. This is shown in subfigure (f) of Figure 1, where values within red circles (i.e., moving objects) are 0. When the flux change percentage in overlapping regions of moving objects exceeds threshold $T$, the threshold map can deactivate strong correlations in these regions, effectively weakening correlation to highlight moving celestial bodies.

Based on the obtained correlation map and threshold map, assuming we have input and reference images $I$ and $R$ with bad and saturated pixels removed, our algorithm proceeds as follows:

**Step 1:** $I$ and $R$ each have their background values subtracted and are normalized to obtain $I_0$ and $R_0$:

$$I_{nb} = I - bkg(I)$$
$$S_I = \sum |I_{nb}|$$
$$I_0 = I_{nb}/S_I$$

$$R_{nb} = R - bkg(R)$$
$$S_R = \sum |R_{nb}|$$
$$R_0 = R_{nb}/S_R$$

Here, $bkg()$ denotes background value calculation. We use the Photutils software developed by Bradley et al. for background estimation, though other methods

may also be employed. The subscript $nb$ indicates background removal, $S_I$ and $S_R$ represent normalization scaling factors, and $\sum abs()$ denotes the sum of absolute pixel values. Note that removing bad and saturated pixels ensures correct normalization.

**Step 2:** Identify pixel intensity differences $D_0$ between $I_0$ and $R_0$:

$$D_0 = I_0 - R_0$$

**Step 3:** Eliminate portions with similar flux distributions while preserving dissimilar portions:

$$P_0 = pearson(I_0, R_0)$$
$$F_0 = diff(I_0, R_0)$$
$$D_1 = D_0 \times (1 - P_0 \times F_0)$$
$$D = D_1 \times S_I$$

Here, $pearson()$ computes the correlation map $P_0$ (modified by equation (4)), and $diff()$ computes the threshold map $F_0$. Equation (16) eliminates similar flux distributions while preserving dissimilar ones, using the threshold map to deactivate strong correlations in overlapping regions of moving objects. Since $I_0$ was normalized in Step 1, we restore it by multiplying by factor $S_I$ to obtain $D$ (see equation (17)). This $D$ is the final processed subtraction image from our algorithm, referred to as the esapd subtraction image.

## 2 Algorithm Performance

To validate our algorithm's performance, we selected three codes for comparison. Among Alard-based implementations (ISIS2.2, DIAPL, and HOTPANTS), we chose HOTPANTS as representative due to its convenient interface. The OIS package provides Bramich-based code, while properimage implements Zackay et al.' s method. We refer to these three algorithms as Alard, Bramich, and properimage. Notably, esapd, Bramich, and Alard ran on a 2020 MacBook Pro M1. Due to installation constraints, properimage ran on a desktop computer (Intel Core i7-10700 CPU @ 2.90GHz) approximately 1.25 times slower than the MacBook Pro M1. When comparing runtime across all four algorithms, we multiplied properimage's execution time by 0.8 for accurate relative comparison. The following experiments demonstrate our algorithm' s performance.

### 2.1 Detection of Moving Objects

We tested images of Apophis captured on February 5, 2013, using the 2.4-meter telescope at the Yunnan Astronomical Observatory' s Lijiang

station. Due to Apophis' s rapid motion, our esapd image subtraction technique clearly reveals its movement. We conducted experiments using cropped 900$\times$900$images.Forproperimageand HOTPANTS, weuseddefaultparameters.Bramichemployeda(21,21)k kerneland15\times\$15$ d-kernel. Since our goal was to detect the moving object Apophis, we set threshold $T$ to a relatively large value of 0.5 to maximize residual elimination and remove false objects.

As shown in Figure 3 [Figure 3: see original paper], the subtraction image produced by esapd is exceptionally clean. In the esapd subtraction image, white points represent the moving object Apophis from the input image, while black points represent it from the reference image. The moving object can be easily identified.

However, the three subtraction images generated by properimage, Bramich, and Alard are not clean, making moving object detection challenging. Pixel value examination using SAOImageDS9 reveals that esapd achieves exceptionally clean subtraction, while other methods cannot. Large-amplitude residuals interfere with Apophis detection. Subtraction artifacts from Bramich and Alard make it difficult to confirm whether a transient source is real—an issue attributed by Zackay et al. to asymmetry in the Alard and Lupton family of methods. With our algorithm, we need only use the DAOFIND program to search for moving objects, whereas the other three algorithms struggle to do so reliably.

Detected Apophis positions are marked with red circles in the esapd subtraction image. Notably, a small black spot appears on the right side of the reference image (within the green circle), caused by a CCD defect in the telescope. In fact, this black spot' s value is only about 200 counts below the image background value of 9850.

Visual inspection reveals that only properimage can detect this black spot, demonstrating its superiority in certain aspects, particularly detail detection. Overall, however, our algorithm exhibits good robustness and the fastest execution speed. The Bramich algorithm runs very slowly; when we adjusted the kernel size to (31,31), execution time reached 191.744 seconds.

To verify the positional accuracy of moving objects detected by our algorithm, we measured positional deviations between targets in subtraction images and their counterparts in input images. For example, we measured the white point within the red circle in the esapd subtraction image from Figure 3 and its corresponding position in the input image, then calculated their positional deviation. We repeated this process for other algorithms. Using 74 Apophis images captured on February 4, 2013, with the Lijiang 2.4-meter telescope across five fields of view, we grouped them by field and selected the final image from each group as the reference, using the remainder as input images. After subtraction, we measured moving object positions using a two-dimensional Gaussian fitting algorithm. Figure 4 [Figure 4: see original paper] presents our results, showing 69 positional deviations for each algorithm. Surprisingly, all four algorithms produce consistent mean values, standard deviations, and offsets. Larger po-

sitional deviations occur due to object overlap. Our esapd algorithm achieves the smallest standard deviation in both x and y axes: mean = 0.0572 and standard deviation = 0.1811 in the x-axis; mean = 0.0498 and standard deviation = 0.1794 in the y-axis. This demonstrates that our esapd algorithm yields relatively stable target positions.

### 2.2 Runtime Comparison

We compared execution times across all four algorithms. We downloaded M35 cluster images from ZTF (Zwicky Transient Facility), cropped them to sizes of 256$\times$256, 512$\times$512, ..., 3072$\times$3072, and measured algorithm runtime in seconds. Figure 5 [Figure 5: see original paper] shows that our esapd algorithm is the fastest, while Bramich is the slowest. For very large images, Bramich may even fail to produce a solution.

Background removal and normalization are two critical operations; without them, our algorithm cannot function. Normalization primarily addresses issues with images having different exposure times. Note that our code does not provide image registration or background estimation algorithms. We use the astroalign algorithm developed by Beroiz et al. for image alignment and the Photutils software by Bradley et al. for 1-D or 2-D background estimation, though other algorithms may also be employed.

Alignment accuracy affects correlation coefficients. We generated a $14\times14 simulated star image containing a single star centered in the image using Photutils (reference image), then$ images with star centers offset from the image center in 0.1-pixel steps (target images). We computed correlation maps between each target image and the reference image, then averaged all pixel values in each correlation map to obtain the correlation coefficient at each offset. As shown in Figure 6 [Figure 6: see original paper], when offsets are (0,0), the correlation coefficient reaches its maximum value of 1; when offsets are (2.5,2.5), the coefficient is approximately 0.85. Experiments show that larger offsets (i.e., lower alignment precision) produce smaller correlation coefficients (without sigmoid modification), while smaller offsets yield larger coefficients. The relationship between correlation coefficient and offset resembles a Gaussian distribution, demonstrating that higher alignment precision produces more accurate correlation coefficients.

### 3.2 Kernel and Threshold Settings and Algorithm Applicability

The p-kernel size relates to the difference in full width at half maximum (FWHM) between the two images: larger FWHM differences require larger p-kernels, and vice versa. For example, for images captured on the same night with similar FWHM, one can set the p-kernel to its minimum size of 3$\times$3. The d-kernel relates to the average FWHM of the two images: larger average FWHM requires larger d-kernels, and vice versa.

Practical experience shows optimal results are achieved when the p-kernel is 2-3 times the FWHM difference and the d-kernel is 2-3 times

the average FWHM, yielding clean subtraction while highlighting moving objects. For instance, when the FWHM difference is 2.36 pixels, a $5\$\times 5p - kernelisappropriate; whentheaverageFWHMis7.08pixels, a15\times\$15$ d-kernel works well. This conclusion was drawn from simulated images; when processing real images, threshold $T$ must also be adjusted to find optimal parameters.

A smaller threshold $T$ provides stronger ability to highlight differences between images, but excessively small values are not ideal. If $T$ is too small, the subtraction image exhibits large-amplitude residuals and even false objects because correlation becomes ineffective and the algorithm degenerates into direct image subtraction. Typically, $T$ is set to 0.1. If the two images differ substantially and one wishes to detect moving objects while maintaining clean subtraction, $T$ can be set larger, such as 0.5, which can be understood as displaying differences between images when flux changes in a region exceed 50%, regardless of correlation strength.

Due to our algorithm' s specific characteristics, its performance is not ideal when point spread functions (PSFs) differ significantly between images—for example, when one has a circular Gaussian PSF and the other has an inclined elliptical Gaussian PSF. However, if both images have elliptical Gaussian PSFs with the same inclination angle, the algorithm' s performance remains largely unaffected. The algorithm also performs poorly on images with FWHM differences exceeding 2 pixels; we recommend using images with FWHM differences within 2 pixels.

### 3.3 Integral Images

To accelerate computation of Pearson correlation coefficients and differential flux coefficients, we adapted Viola and Jones' s integral image method for rapidly calculating Haar-like features. Formulas (2) and (5) contain numerous summation operations such as $\sum i$, $\sum r$, $\sum i^2$, $\sum r^2$, and $\sum ir$. We employ integral images to accelerate these calculations. Using integral images not only speeds up computation of correlation and differential flux coefficients but also ensures that runtime remains unchanged when p-kernel or d-kernel sizes vary, making our algorithm' s time complexity independent of parameters.

### 3.4 Noise Estimation

Noise in subtraction images originates from two sources: noise in the input image and noise in the reference image. Assuming Poisson-distributed noise in both images, we derived a noise estimation formula for subtraction images:

$$\sigma_D^2 = S_I^2(R + I)$$

Note that this formula applies only when $F_{cc}(r, i) = 0$ or when correlation is very weak (Pearson correlation coefficient $< 0.2$).

## 4 Summary and Outlook

This paper proposes an alternative image subtraction algorithm, esapd, designed for moving object detection. The algorithm aims to eliminate portions with similar flux distributions between two images while preserving dissimilar portions based on correlation. Experiments demonstrate that the algorithm can rapidly identify differences between images and search for moving objects. However, limitations exist: when PSF distributions differ substantially between images, the algorithm' s performance requires improvement. Future work will incorporate PSF information to further investigate astronomical image subtraction techniques.

We thank the staff of the Lijiang 2.4m telescope for their support and Guo Bifeng for assistance with manuscript preparation.

## References

[1] CROTTS A P S, TOMANEY A B. Results from a survey of gravitational microlensing toward M31[J]. The Astrophysical Journal, 1996, 473(2): L87.
[2] KOCHANSKI G P, TYSON J A, FISCHER P. Flickering faint galaxies: few and far between[J]. The Astronomical Journal, 1996, 111: 1444.
[3] ALARD C, LUPTON R H. A method for optimal image subtraction[J]. The Astrophysical Journal, 1998, 503(1): 325-331.
[4] ALARD C. Image subtraction using a space-varying kernel[J]. Astronomy and Astrophysics Supplement Series, 2000, 144(2): 363-370.
[5] WOZNIAK P R. Difference image analysis of the OGLE-II bulge data. I. The method[J]. Acta Astronomica, 2000, 50: 421-450.
[6] BRAMICH D M. A new algorithm for difference image analysis[J]. Monthly Notices of the Royal Astronomical Society: Letters, 2008, 386(1): L77-L81.
[7] YUAN F, AKERLOF C W. Astronomical image subtraction by cross-convolution[J]. The Astrophysical Journal, 2008, 677(1): 808-812.
[8] ZACKAY B, OFEK E O, GAL-YAM A. Proper image subtraction—optimal transient detection, photometry, and hypothesis testing[J]. The Astrophysical Journal, 2016, 830(1): 27.
[9] BRADLEY L, et al. Astropy/photutils: 1.1.0 [CP/OL]. [2021-03-20]. https://zenodo.org/record/4624996.
[10] JOYE, W. SAOImageDS9/SAOImageDS9 v8.0.1 [CP/OL]. [2019-01-03]. https://zenodo.org/record/2530958.
[11] STETSON P B. DAOPHOT: A computer program for crowded-field stellar photometry[J]. Publications of the Astronomical Society of the Pacific, 1987, 99(613): 191.
[12] PENG H W, PENG Q Y, WANG N. Precise CCD positions of Himalia using Gaia DR1 in 2015–2016[J]. Monthly Notices of the Royal Astronomical Society, 2017, 467(2): 2266-2273.
[13] BEROIZ M, CABRAL J B, SANCHEZ B. Astroalign: A Python module for astronomical image registration[J]. Astronomy and Computing, 2020, 32: 100384.

[14] VIOLA P, JONES M. Rapid object detection using a boosted cascade of simple features[C]//Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Ieee, 2001, 1: I-I.

*Note: Figure translations are in progress. See original paper for figures.*