

A New Approach for Detection of Moving Objects in FITS Images: A-Track

T. Atay^a, M. Kaplan^a, Y. Kılıç^a, N. Karapınar^a

^a*Akdeniz Üniversitesi, Fen Fakültesi, Uzay Bilimleri ve Teknolojileri Bölümü, Antalya,
Türkiye*

Abstract

We have developed an open-source, cross-platform pipeline for the detection of moving objects such as asteroids or comets in sequential telescope images. We tested the pipeline using the TÜBİTAK National Observatory data which were acquired by a 1.0 meter diameter telescope (T100) and an SI CCD 1100 camera. We found that our pipeline is at least as successful as any other software used for asteroid detection, in terms of detection efficiency, stability, and processing time.

Program summary

Program title: A-TRACK

Catalogue identifier: a-track_v1_0

Program summary URL: <https://github.com/akdeniz-uzay/A-Track>

Program obtainable from: <https://github.com/akdeniz-uzay/A-Track/archive/master.zip>

Licensing provisions: Standard GNU General Public Licence, <http://www.gnu.org/licenses/gpl-3.0.en.html>

No. of lines in distributed program, including test data, etc.:

No. of bytes in distributed program, including test data, etc.:

Distribution format: .zip

Programming language: Python.

Computer: Personal Computer.

Operating system: Any OS where Python are installed.

RAM: Around ... Mbytes

Classification: ...

External routines: numpy ...

Subprograms used:

Nature of problem:

Solution method:

Email address: tolgapd@gmail.com (T. Atay)

Restrictions:

Running time:

Keywords: methods: data analysis, minor planets, asteroids, techniques:
image processing

2015 MSC: 10-13, 99-00

1. Introduction

Asteroids and comets offer valuable information about the formation and evolution of our Solar System. But there is also another reason for the detection and observation of asteroids and comets, near-Earth objects (NEO) in particular, that Earth may be impacted by one of them. Over long periods of time, this possibility is not negligible. With our current technology, it may be possible to deflect a threatening asteroid or comet away from Earth, given enough warning time. This is why many research groups from all over the world contribute their best efforts to detect and track these objects.

Most of these groups use robotized telescopes with high-resolution cameras and optical systems to scan the sky all night long. However, none of these groups use an open-source/free software that detects asteroids and comets automatically. Some independent researchers prefer commercial software such as Astrometrica (Raab, 2012) or CoLiTec (Savanevich et al., 2012). These are relatively easy to use, yet inadequate for large archives as they need human interaction.

In this work, we introduce a new method to detect moving objects in sequential telescope images in FITS format. The method needs at least 3 sequential images of the same sky region taken one after the other on the same night. We used this method to develop an open-source (licensed under GPL v3), cross-platform pipeline: A-Track. It is easy to use, it is fast, and it doesn't need human interaction. We used Python for the coding since it is easy to understand and has many astronomical modules to work with. Our software has been tested on GNU/Linux (Fedora 20, Ubuntu 14.10) and Mac OS X (10.10.4,

Yosemite). The workflow of the pipeline, the packages and the modules used, and details about the developing stage are given in section 2. In section 4, we present some test results, moving objects detected by A-Track on different data sets.

2. Automatic Detection of Moving Objects

2.1. Main Components

In principle, the moving objects in sequential CCD images can be identified by tracking their motion with respect to the stationary objects (stars). One requirement is that the moving object is detected by the telescope/CCD system in most (if not all) of the images. Another requirement is that the moving object is fast enough that its motion from image to image is detectable.

A-Track uses the following steps for moving object detection:

1. Goruntularin Malte Tewesin alipy (Tewes, 2015) modulu yardimiyla hizalanmasi
2. SExtractor (Bertin and Arnouts, 1996) ile tum goruntuler uzerindeki nesnelerin tanimlanmasi ve katalog dosyalarinin olusturulmasi
3. Tum goruntulerden olusturulan katalog dosyalarinin tek bir master dosya icerisinde toplanmasi
4. Her bir katalog dosyasi icerisindeki her bir nesnenin master katalog dosyasi ile karsilastirilmesi ve kiyaslanan dosyanin hareketli nesne olup olmadiginin sorgulanmasi (ilgilenilen nesneden ne kadar var? Eger 1 tane ise hareketli nesne olmaya aday demektir) ve hareketli nesne aday kataloglarinin olusturulmasi
5. Gelistirilmis Dogru Tanimlama algoritmasinin uygulanmasi ve dogrularin etiketlenmesi
6. Gorsellestirme ve hareketli goruntu haline getirilmesi
7. Kullaniciya sonucun bildirilmesi
8. Astrometrik cozumun uygulanmasi (hedef)

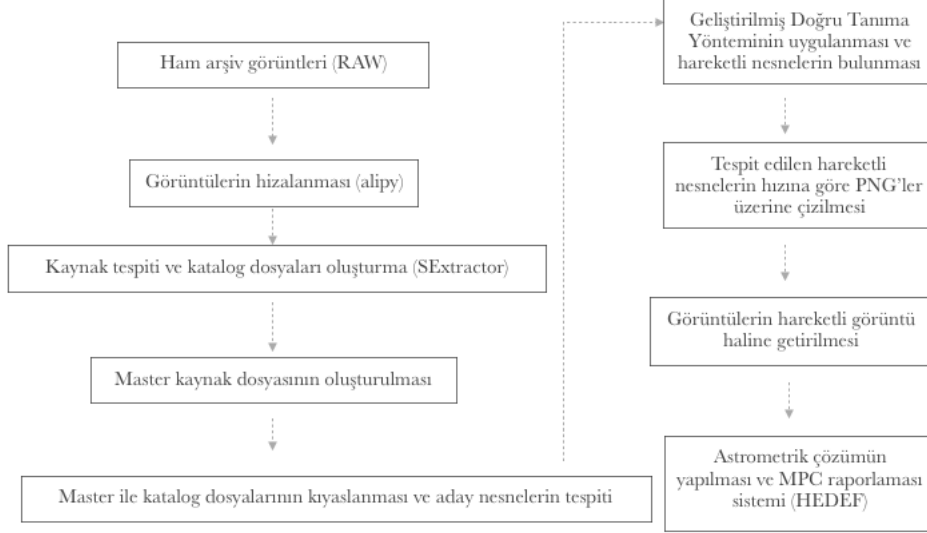


Figure 1: Pipeline'nin algoritması

9. Sonuclarin MPCye raporlanmasi (Fig. 1)

Tabii goruldugu gibi bu adimlarin hepsi ciddi miktarda donanim (processing time) gerektirdiginden ozellikle hareketli aday nesnelerin tespiti ve *Gelistirilmis Dogru Tanimlama* adimlarinda islemleri paralellestirdik. Butun bu islemlerin detayli anlatimi bir alt baslikta verilmistir.

2.2. Goruntulerin Secilmesi

Pipelineda kullanılan veri setleri 2500 m. yukseklikteki TUBITAK Ulusal Gozlemevinde (TUG) bulunan 100 cm capli T100 teleskobuna bagli Spectral Instruments (SI) 1100 4096 x 4037 px, CCDsinden elde edilmistir. Yazilimin performansini denetleyebilmek adina veri setlerinin SNR degerlerinin dusuk, orta ve yuksek olmasina ve kalabalik alanli gozlem verilerin secilmesine ozen gosterilmistir. Secilen goruntulerde algoritmamizin isletilmesinde son derece onemli olan gozlem zamani (DATE-OBS), alinan goruntunun poz suresi (EXPTIME), binning miktarı (XBIN, YBIN), boyutu (NAXIS1, NAXIS2) ile ilgili anahtarların goruntu basliklarında olması gerekmektedir.

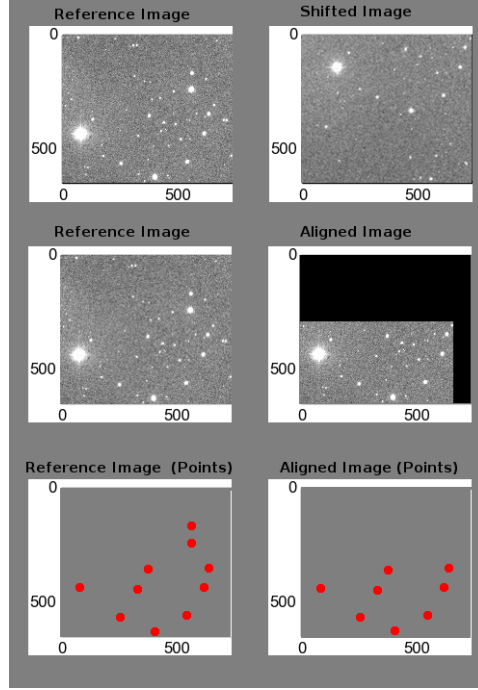


Figure 2: Hizalama adimi

2.3. Goruntulerin Hizalanmasi

Hizalama adimi ardisek gelen goruntulerdeki hareket eden nesnenin tepiti acisindan onemli bir adimdir. Biz hizalama islemini gerceklestirmek icin, Python programlama dili ile gelistirilmis ve GPL ile dagitilan Malte Tewesin alipy adli yazilimini kullandik. Alipyda hizalama islemleri icin, elde ettigi referans katalog dosyasindaki oruntu ile diger goruntulerin katalog dosyasindaki oruntuyu eslestirmek icin (Lang et al., 2010) calismasini temel alarak scipy afin donusunu veya PyRAFin geomap/gregister modulunu kullanmaktadir. Bizim icin sadece afin donusumu yeterli olmaktadır. Bu islemin nasil gerceklestiginin basit bir anlatimini Fig. 2te inceleyebilirsiniz.

2.4. Hizalanan Goruntulerden Nesne Tespiti

Gelistirdigimiz pipeline’de hizalama komutu (unless `-skip-align`) calistirildiktan sonra X , Y shift ve θ ’ya gore hizalanan goruntuler belirtilen dizin iceri-

sine kayıt edilir (atrack/). Artık yüksek hassasiyetle hizalanmış görüntülerden nesne tespiti amacımıza uygun hale gelmiştir. CCD görüntülerinden kaynak tespiti yapabilen pek çok yazılım vardır. Bunlara örnek verecek olursak; astrometry.netin image2xy modulu, IRAFin daofind ve starfind komutları ve buna benzer pek çok IDL kutuphanesi bulunmaktadır. Fakat, CCD görüntülerden kaynak tespiti yapmak isteyince şüphesiz bu konuda hem çok başarılı ve de hem de çok hızlı olması bakımından ilk akla gelen yazılım SExtractor'dur (Bertin and Arnouts, 1996). Biz bu çalışmada SExtractorun en son sürümü olan 2.19.5 sürümünü kullandık. Pipelineda daha sonrak cisimlerin tanımlanması, daha fazla hareketli nesnenin tespit edilmesine olanak sağlayacaktır. Dolayısıyla fonksiyonların yazılmasında SExtractor'u kullanırken çeşitli tespit parametrelerinin kullanıcıya bağlı olmasına özellikle özen gösterdik. Görüntülerden nesne tespitini N. Cantalein pysex modulu aracılığı ile SExtractor'u kullanarak, daha sonra aday nesnelerin karakteristiklerini incelemek adına, nesnelerin tespit loglarını (FLAGS), nesnelerin görüntü üzerindeki konumlarını (XIMAGE, YIMAGE), akılarını ve hatasını (FLUXAUTO, FLUXERRAUTO), arkaplan değerini (BACKGROUND) ve basıklık değerini (ELONGATION) her bir nesne için kendi görüntü dosyasıyla ilişkilendirerek katalogladık (SExtractor output).

2.5. Master Katalog Dosyası ve Aday Nesnelerin Belirlenmesi

Her bir görüntü için nesne tespiti sonrası oluşan katalog dosyalarından (*.pysexcat) hareketli nesne adaylarının belirli bir tolerans değerine göre ayrılması gerekmektedir. Bu ayırım sonrası hareketsiz olan nesnelerden arda kalanlar yalnızca; SEEINGden dolayı merkezi tolerans değerinden fazla oynamış hareketsiz nesneler (cogunlukla yıldız), kozmik isinlardan dolayı asiri yüklenmiş bölgeler ve gerçek hareketli nesneler olacaktır. Bu işlemin gerçekleşmesi için kurguladığımız algoritma gereği, ilk olarak bir Master Katalog Dosyasına (MKD, master.pysexcat) ihtiyaç duyulmaktadır. MKD dosyası aslında tüm görüntülerden oluşturulmuş katalog dosyalarının hepsini bir katalog dosyasında birleştirilmiş halidir. Böylelikle her bir görüntüdeki her bir nesne MKD içinde belirli bir tol-

erans degerinden uzak olmamak kosulu ile, kac kez tekrar ediyor kontrol edilir ve eger bu kontrol sonrasinda tolerans icerisinde kalan nesne sayisi 1 adet ise nesnenin hareket ediyor olma olasiligi yuksektir ve Hareketli Nesnelerin Aday Kataloguna (*HNAK*) kaydedilir (*.cnd). Bu tekrar islemi 1den fazla ise o bir sabit nesne demektir ve *HNAK*ya kaydedilmez. Bu islem her bir katalog dosyasinda bulunan her bir nesne icin gerceklestirilir. Islem bittiginde diger bir katalog dosyasina gecilir. Bu islemin matematiksel olarak anlatimi Denklem 1de gosterilmistir.

$$\sum_{n=0}^{n_{\max}} \sum_{m=0}^{m_{\max}} \sqrt{(m_x - n_x)^2 + (m_y - n_y)^2} \leq \delta x \quad (1)$$

Denklem 1de katalog dosyasi icerisindeki nesneler (n) sirasiyla master dosyasindaki nesnelere (m) uzaklik bakimindan kiyaslaniyorlar. Master dosyasindaki bu kiyaslamanin sonucunda hareketli nesnelerin yalnızca bir kez bu sarti saglamasi gerekir. Tabii bu ornegi aciklamak icin sadece MKD ile kiyaslama esnasinda yalnızca nesnenin toleranstan uzaga kacabilme sarti degerlendirilmemistir. SExtractor'dan nesnelerin karakteristigi ile ilgili farkli degerler de alindigindan bolum 2.3'te bahsetmistik. Eger sadece Denklem 1'deki uzaklik sartini aday nesne belirlemek icin kullanirsak; pek cok gorus kalitesinden, kozmik isinlardan ve arkaplan dalgalanmalarından kaynakli nesnelerin de aday nesne kataloguna alinmasina sebep olabiliriz. Bu yuzden istenmeyen bu nesneleri SExtractor'den gelen diger ozelliklerini de dikkate alarak elemek durumundayiz. Bunlardan en onemlileri tabii ki de HNAK'a atilacak nesnelerin saglamasi gereken maksimum FWHM ve minimum FWHM degerleridir. Pipeline'de bu minimum FWHM degeri 1 px olarak belirlenirken, maksimum FWHM degeri MKD'deki tum nesnelerin FWHM degerinin medyaninin 2.5 (optional) kati olarak belirlenmistir. Boylelikle SEEING'den bagimsiz olarak hareketli nesneler rahatlikla belirlenebilecektir. Ayrica, her bir nesnenin SNR degeri 2.5 (optional)'dan buyuk olmak durumundadir. Bunlarin yani sira ayrica da SExtractor katalog dosyalarinda bulunan nesnelerin FLAG degerleri, basikligi (ELONGATION) da HNAK'a nesne atilirken degerlendirilmektedir. Tum bu kiyaslamalar GitHub uzerinden pay-

lastigimiz kaynak kodda mevcuttur (*asteroids.py, detect_candidates*).

2.6. Astronomik Amaçlı Gelistirilmiş Doğru Tanıma Yöntemi ve Nesne Etiketleme

Literatur incelendiğinde farklı doğru tanıma algoritmaları bulunmaktadır. Görüntü işlemede bunlardan en çok kullanılanı Hough dönüşümü ve bu dönüşümün varyasyonlarıdır ((Hough, 1962), (Duda and Hart, 1972)). Fakat bizim çalışmamıza entegre edilebildiği açısından (Chen and Chung, 2001)'in çalışmasıyla ortaya konulmuş Doğru Tanıma Yöntemi (DTY)'nin en uygun yöntem olduğunu düşünmekteyiz. Görüntü içerisinde doğru oluşturmuş noktaların belirlenebilmesi sebebiyle öncelikle DTY tercih edilmiş ve amacımıza uygun olarak geliştirilmiştir. DTY, ilgilenilen görüntüdeki (ana görüntü deil miydi?) kaynak noktalarına (koordinatlara) uyarlandığında u ekilde çalışır (Fig. 2). Önce iki nokta ele alınır, sonra üçüncü bir nokta alınıp, bu iki nokta ile doğruda olup olmadığı kontrol edilir. Eğer alınan üç noktanın doğruda olduğu doğrulanırsa bu doğrulturma ana görüntüdeki diğer noktalar için de tekrarlanır. Burada zor olan, üç noktanın doğruda olma kriterlerinin belirlenmesidir. Kullanılan yöntemde üç noktanın doğruda olmasının birinci koulusu, her bir nokta (v_i) ve bu noktaya karılık gelen koordinatlar $v_i=(x_i, y_i)$ olmak üzere, bu üç noktanın oluşturacağı üçgenin alanının değerinin sıfır veya sıfıra çok yakın olmasıdır. Bu değerleri, üç noktası bilinen üçgenin alanı formülünden hesaplayabiliriz (denklem 2).

$$|z| = \frac{1}{2} |(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)| \quad (2)$$

Dijital görüntülerde noktaların piksellerden Fig. 3 farklı görüntüden seçilmiş hareketli nesne adaylarının uzayda sanal doğru üzerinde oluşturduğu üçgenin DTY'ye göre semasi oluştuğu göz önüne alındığında, bu değerlerin tam olarak sıfır olmasa da sıfıra yakın olması beklenir. Ancak bu alanın sıfıra yakın olduğu her durumda, kullanılan üç noktanın doğruda olduğu söylenemez. Üç nokta birbirine çok yakın olduğunda da alan sıfıra yakın olabilir. Dolayısıyla ikinci bir kriter ihtiyacı vardır. Eğer seçilen üç nokta doğrudası ise bu noktalardan her birinin diğer ikisinden geçen doğruya uzaklığı (oluşan üçgenin seçilen noktasından karşı kenara inen yüksekliği) da sıfıra yakın olmalıdır. i, j, k noktalarından her

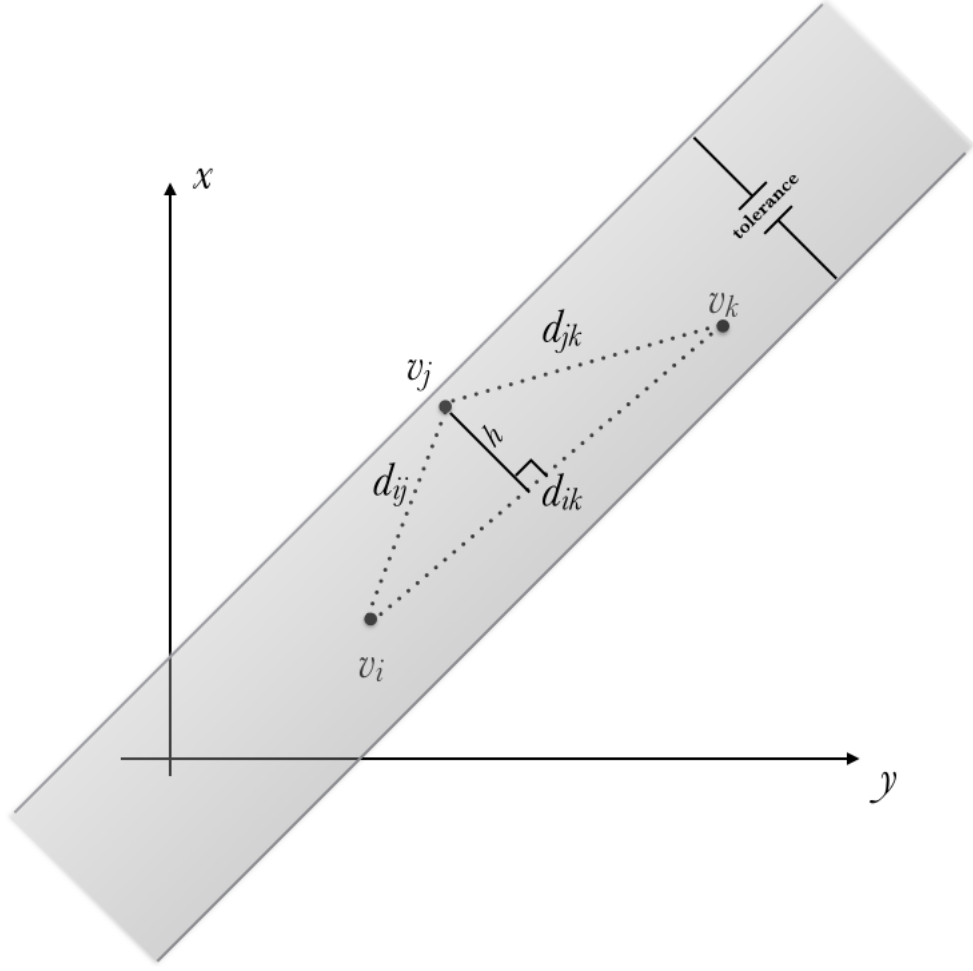


Figure 3: Uc farkli goruntuden secilmis hareketli nesne adaylarinin uzayda sanal dogru uzerinde olusturdugu ucgenin DTY'ye gore semasi

biri icin o noktaya karsilik gelen $d_{i \rightarrow jk}$ yuksekligi bir noktanin bir dogruya olan uzakligi formulu kullanilarak hesaplanir ve en kucuk yukseklik degeri belirlenir (denklem 3).

$$d_{k \rightarrow ij} = \frac{(x_j - x_i) y_k + (y_i - y_j) x_k + x_i y_j - x_j y_i}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \quad (3)$$

Eger hesaplanan yukseklik degeri belirli bir tolerans araliginda ise, ucuncu

bir kriter olarak, bu uc noktanin birbirlerinden belirli bir eik degeri kadar uzak olmalari istenir (ucgenin tabani). Bu uc kriterin hepsini saglayan noktalar dogruda kabul edilir. Algoritma, ana goruntudeki kaynak noktalarinin (coklu goruntulerden elde edilen potansiyel hareketli cisimlerin koordinatlarinin) dogrudas olup olmadiklarini yukaridaki uc kritere gore degerlendirdikten sonra, bu kaynak noktalarin ayni dogru uzerinde olup olmadigina karar verir (Chen and Chung, 2001).

(Chen and Chung, 2001)'in uyguladigi bu yontem tek bir goruntu uzerindeki surekli noktalar (edges) bulma amaci tasiyordu. Bu yuzden noktalarin arasindaki uzakligin belirlenen limitlerin disinda farkli bir parametre ile denetlenmesinin geregi yoktu. Fakat bizler uzayda hareketli nesneler aradigimiz icin ve bu nesneler belirli bir oz hareket ile CCD uzerinde hareket edeceginden, Fig. 3'te de gosterilmis d_{ij} , d_{jk} , d_{ik} dogrularinin vektorel buyukluklerinin birbirleri ile orantili olmasi gerekmektedir. Bu calismada boyle bir kontrol mekanizmasi DTY'den once kullanilmistir. Bu ek kontrol mekanizmasi su sekilde calismaktadir. Dusuk plak eseline sahip CCD goruntusu uzerinde asteroidler mukemmel bir dogru uzerinde yol alirlarlar. Bu yer degistirme de zamana mutlaka bagimlidir. Bu sebeple $\frac{d_{ij}}{t_1} = \frac{d_{jk}}{t_1}$ olur. Tabii ki secilen ilk iki nokta ve sonrasinda secilecek ucuncu nokta herhangi ikili nokta cifti olmamalidir. Eger butun noktalarin kombinasyonu olacak sekilde secim yapilrsa ayni dogru uzerinde bulunan ve birbirinden cok uzakta bulunan uclu nokta gruplari da hareketli nesne olarak algilanabilir ve gereksiz bir islem yuku ile karsi karsiya kaliriz. Bunun icin ilk olarak bir baslangic parametresine ihtiyac duyulmaktadır (denklem 4).

$$r = (t_{12} - t_{23}) \cdot \frac{V_{\max}}{p \cdot b} \quad (4)$$

Denklem 4'te p: piksel olcegini, b: verinin binning degerini temsil etmektedir. Hesaplanacak baslangic parametresi, secilecek olan ilk iki nokta arasindaki uzakligin ne olmasi sorusuyla alakalidir. Burada V_{\max} a herhangi bir deger verilemez. Bu deger arastirilan hareketli nesnelerin turu ve yorungesiyle alakalidir. Bizler bu calismamizda, bu baslangic parametresini Trojan'larin oz hareketleri

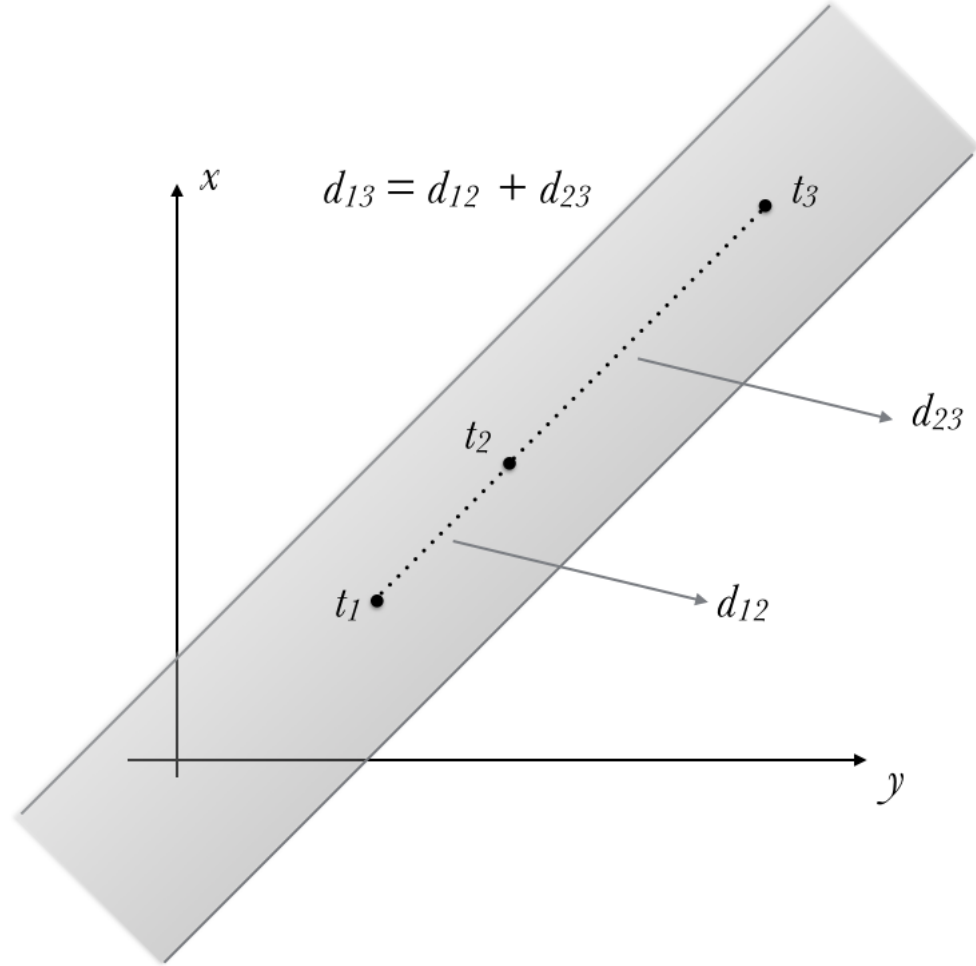


Figure 4: Sabit hızlı hareketli nesnenin farklı t zamanındaki konumları

icin çok hızlı sayılabilecek bir değer olan ve aynı zamanda Aten'ler için de ortalama hızlı kabul edilebilir bir öz hareket değeri olan $0.03 /s$ (V_{max}) değerini kullandık. Tabii bu değeri belirlememizde kendi arşivimizden tespit edilen ve yakalayabildiğimiz hızlı asteroidlerin öz hareketleri de etkili olmuştur. Üçüncü nokta için ise böyle bir kıyaslamaya gerek yoktur. Çünkü üçüncü noktanın Fig. 4 Sabit hızlı hareketli nesnenin farklı t zamanındaki konumları seçimi tamamen ilk iki nokta arasındaki hıza bağlıdır. Bu durumu daha iyi anlatması bakımından Fig. 4 ve Denklem 4 incelenebilir.

$$d_{23} = \frac{t_3 - t_2}{t_2 - t_1} \cdot d_{12} \quad (5)$$

Denklem 4 kurgulamasinin aslında asteroidlerin gözlem süresi boyunca hızlarının sabit olduğu varsayımından ortaya çıktığından bahsetmistik ($V_{asteroid} = \frac{d_{12}}{t_{12}} = \frac{d_{23}}{t_2}$). CCD görüntülerinin piksellerden (yani kesikli bir iki boyuttan) olustugunun ve de nesnelerin merkezlerinin ardışık her görüntüde -kötü seeing ya da düşük SNR- den dolayı mükemmel olarak belirlenememesinden kaynaklanan konumdaki hatalar da göz önünde bulundurulursa, üçüncü noktanın konumunun (d_{23} uzunluğu da denebilir) kesin olarak belirlenmesini biraz zora sokmaktadır. Bu yüzden üçüncü noktanın aranması gereken konum aralığı için belirli bir tolerans değerinin verilmesi (biz 1 px kullandık) son derece akıllıca olacaktır. Bu durum yazılımsal olarakta incelemek istenirse *asteroids.py* modulünde *detect_candidates* fonksiyonu incelenebilir.

DTY'nin temel şartları ile birlikte astronomik hız ve konum şartlarını da sağlayan uçuş nesneleri bir dizi içerisinde biriktirilir. Bu uçuş noktalarının birbirlerinin devami olduğuna karar verilir (collectpointsonline fonksiyonu), daha sonra birer doğru numarası verilerek Fig. 5'da olduğu gibi nesnelerin hesaplanan tüm özellikleriyle birlikte kullanıcıya çıktı verilir.

2.7. Belirlenen Hareketli Nesnelerin Görselleştirilmesi ve Bazı Görselleştirme Araçları

Hareketli nesnelerin belirlenmesi ve doğru numaraları ile kullanıcıya gösterilmesinden sonra kullanıcı, tespit edilen bu nesneleri FITS görüntüleri üzerinde görmek isteyebilir ve de hatta animasyon görüntüsü olarak hareketli nesneleri izlemek ihtiyacı duyabilir. Çünkü Fig. 5'da görüldüğü üzere sonuç çıktısında kırmızı ile işaretlenmiş bir takım beklenenin altında hızlara sahip nesneler bulunmaktadır. Bu nesneler pipeline tarafından otomatik olarak bizim belirlediğimiz minimum hareket hızından (0,070 px/min, optional) düşük nesnelerdir ve bu nesnelerin gerçekten hareketli nesne olup olmadığı görsel olarak teyit edilmelidir.

Görselleştirme işlemi sırasında FITS (Flexible Image Transport System) görüntüleri PNG (Portable Network Graphics) görüntülerine *f2n* adlı python mod-

List of Fast Moving Objects

	file_id	flags	x	y	flux	background	lineid	skymotion(px/min)
0	0	3	374.8694	329.1229	127676.10	17124.77	1	0.265184
1	1	3	378.9566	325.7019	67018.31	17244.14	1	0.265184
2	2	0	381.7933	323.1091	50813.62	17356.61	1	0.265184
3	0	3	739.8054	882.1669	31002.35	17941.59	2	0.248053
4	1	0	743.5338	878.6925	46015.26	18032.85	2	0.248053
5	2	0	746.0291	876.2289	55320.64	18135.46	2	0.248053
6	3	0	748.5798	873.8578	67859.38	19127.37	2	0.248053
7	0	0	514.4425	1713.3478	51557.61	17170.30	3	0.190886
8	1	0	518.3229	1714.4528	46155.12	17232.78	3	0.190886
9	2	0	520.7471	1715.1086	52010.42	17289.16	3	0.190886
10	3	0	523.4231	1715.7615	50869.45	18205.70	3	0.190886
11	0	0	1764.4980	1326.9921	22010.26	18220.36	4	0.204244
12	1	0	1767.5900	1329.9253	27741.96	18296.15	4	0.204244
13	2	0	1769.6801	1331.9542	31412.27	18357.71	4	0.204244
14	3	0	1771.6187	1333.9419	22732.86	19315.51	4	0.204244
15	0	0	592.6536	1596.6654	97062.45	17444.11	5	0.073761
16	2	0	598.1887	1593.8529	89803.48	17588.52	5	0.073761
17	3	0	600.3657	1592.6549	85916.45	18529.18	5	0.073761
18	1	0	595.9127	1595.1519	93927.04	17523.44	5	0.073761
19	1	3	1690.9020	598.4935	98529.22	18033.86	7	0.160735
20	2	3	1690.4730	596.0413	93733.27	18162.72	7	0.160735
21	3	3	1690.1195	593.8766	85155.14	19141.18	7	0.160735
22	1	0	1324.9473	812.7097	35222.00	18387.54	8	0.144023
23	2	0	1327.2312	812.7218	31042.21	18482.68	8	0.144023
24	3	0	1329.1425	812.6354	26938.68	19460.28	8	0.144023

List of Slow Moving Objects (Please check these objects! Are these really MOs?)

	file_id	flags	x	y	flux	background	lineid	skymotion(px/min)
0	0	7	1416.9884	1292.7056	342927.30	18325.19	6	0.034303
1	2	7	1417.9171	1292.2036	406497.60	18485.51	6	0.034303
2	3	3	1418.5146	1292.0249	399735.00	19436.07	6	0.034303
3	1	3	1179.0000	1522.0000	98097.62	17987.72	9	0.068382
4	2	3	1178.2284	1523.1299	218389.40	18036.56	9	0.068382
5	3	3	1177.8522	1523.6283	152254.50	18986.80	9	0.068382

Elapsed time: 0 min. 23.32 sec.

Figure 5: Bir gozlem gecesinde rastgele secilmis 4 goruntuden tespit edilmiş hareketli nesneler (Terminal çıktısı)

ulu ile donusturulmaktadır. Bu islem sirasinda tespit edilen nesneler tolerans degerinden buyukse yesil renkte, kucukse kirmizi renkle isaretlenmektedir. Bu-tun goruntulerde bu islemler bittikten sonra istege bagli olarak, donusturulen bu goruntulerden bir animasyon goruntusu imagemagick tarafindan olusturulmak-tadir. Fig. 6'de gorsellestirme islemi sonucunda olusturulmus bir PNG gorun-tusu gosterilmektedir.

2.8. Pipeline'da Coklu Islemci (Parallelestirme) Kullanimi

Gelisen teknolojiyle birlikte, her gecen gun dijitallestirilmiş gorsel astronomik veriler ve veri setleri arxiv üzerinde son derece buyuk alanlar kaplamaktadır.

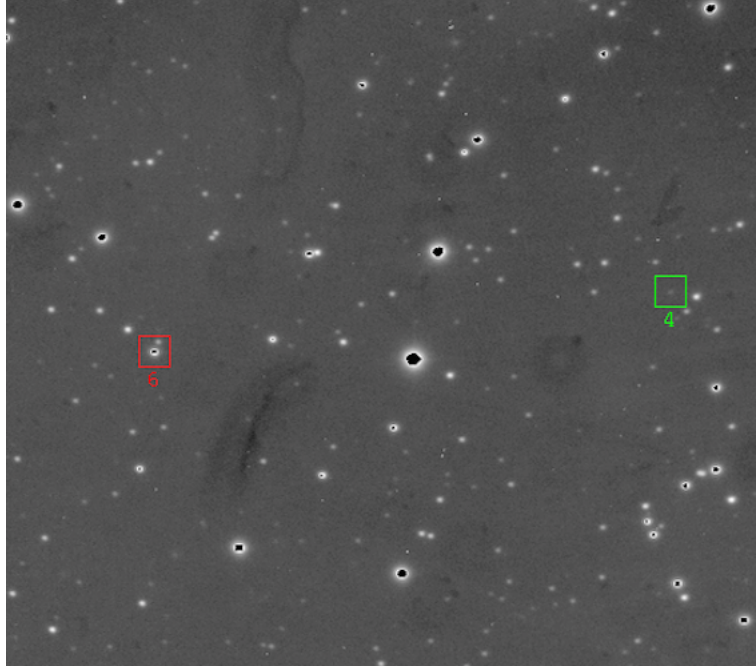


Figure 6: Tespit edilen hareketli nesnelerin hiz kriterine gore farkli renklerde gosterilmesi

Ozellikle gelismis CCD'lerle yapilan gozlemler sonucunda gecelik elde edilen veriler bile Gigabyte'lari hatta Terabyte'lari bulmaktadır. Boylesi buyuk verileri islemek ve analiz etmek icin ciddi islemci hizi ve yuksek hafiza birimlerine ihtiyac duyulmaktadır. Bu donanimlarla birlikte gelistirilen yazilimlarda bu durum goz onunde bulundurularak donanimi etkin ve verimli bir sekilde kullanilmasi adina algoritmanin paralellestirilebilir islemlere zemin saglayacak sekilde gelistirilmesi gerekir. *Bu calisma sadece tek bir gecenin gozlem verileri icinden hareketli nesnelerin bulunmasi amaciyla yapilmamistir. Pipeline'in bir arhive uygulanmasi planlanmaktadır.(bu alma alma/yazlm sadece tek bir gecelik gozlem verileri iinden hareketli nesnelerin bulunmasnda kullanilabilecegi gibi ariv verilerine de uygulanabilecek ekilde tasarlanmtr neriyydi MK)* Bu sebeple *detect_candidates* ve *detect_segments* fonksiyonlarinin tek bir islemciye yuklenilerek calistirilmasi son derece zaman alan bir islem olacaktir. Amacimizin cok sonuk hareketli nesnelerin tespit edilmesi oldugu dusunulurse, SExtractor ile kal-

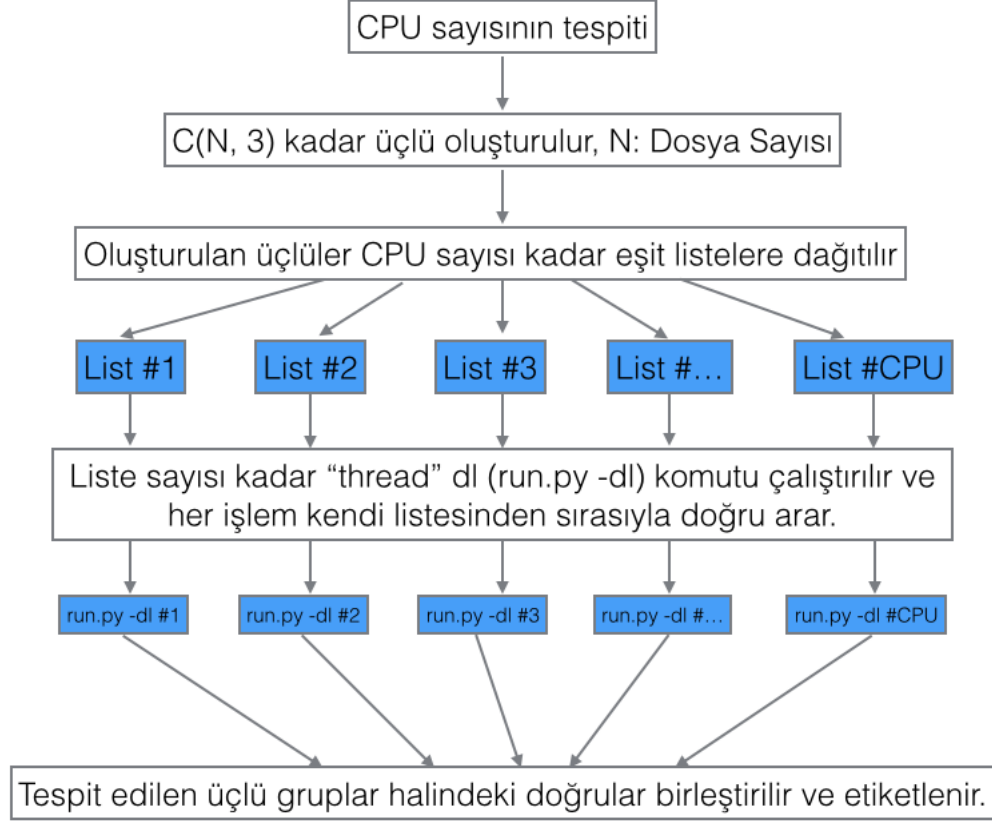


Figure 7: Paralelize edilmiş işlemlerin akis semasi

abalık bölgelerden tanımlanan nesne sayıları binleri hatta onbinleri bulmaktadır. Bu aşamadan sonra *HNAK*'a düşen aday nesneler de yüzleri hatta binleri bulabilmektedir. Ardışık olarak alınmış görüntülerden hareketli nesnenin herhangi bir görüntüde tanımlanamama olasılığına karşın da seçilen 3 nokta sırasıyla $C(N : Dosyasayısı, 3)$ olmaktadır. Yani N değeri arttıkça daha fazla arama işlemi gerçekleştirilmektedir. Bu sebeple hem zamandan tasarruf etmek, hem de donanımı verimli kullanmak adına pipeline'da $C(N, 3)$ kombinasyonlarını tek bir işlemci yerine, $C(N, 3)/CPU\ COUNT$ kadar işlemci ile *detect_{candidates}* ve *detect_{segments}* fonksiyonları çalıştırılmaktadır (Fig. 7).

3. Gelecek Hedefler (Future Works)

Buyuk veri iceren astronomik arxivlerden hareketli nesnelerin, belirlenen algoritmanın geliştirilmesi suretiyle tespiti, bu calisma ile mumkun hale gelmistir. Fakat bu pipeline'i kullanacak kullanıcı yelpazesi genellikle bilim insanlarından olusacaktır. Hal boyle bile olsa kullanıcılara islemlerin mumkun olan en kısa surede bitirilmesi ve de istenilen komutların en kısa surede kolaylıkla verilmesi acısından, bir kullanıcı dostu grafik arayuz (GUI) sunulması da ilerleyen surelerde dusunulmektedir. GUI'nin yani sıra tespit edilen hareketli nesnelerin astrometrik (WCS - World Coordinate System) cozumunun de yapilarak MPC (Minor Planet Center)'ye raporlanması adiminin da eklenmesi orta/ksa vadeli hedeflerimiz arasındadır. Bu hedefler sirasiyla asagidaki gibidir.

- WCS cozumunun tum görüntüler için yapılması
- Tespit edilen nesnelere XY2RADEC donusumunun uygulanması
- MPC raporlama ozelliginin eklenmesi
- Kullanıcı dostu GUI eklenmesi

4. Sonuc ve Tartisma

Gelistirdigimiz yazılım TUG'dan farklı gecelerde alınmis veri setlerine uygulanarak denenmis ve pipeline ile TUG yerleskesinin gorus kalitesi ve $T100$ teleskobunun limit parlakligi ($\approx 21.5^m$) da goz onune alinarak, gozlem yapılan bolgeye ait MPC (MPC katalog taraması) taraması sonucu bulunması gereken (beklenen) asteroidler ile karsilastirma yapilarak hareketli nesneler literatur ile dogrulanmistir. Buna ek olarak MPC veritabanında bulunmayan fakat, yazılımın tespit ettigi hareketli nesneler de tespit edilmistir.

Gelistirilen pipeline; ilgilenilen cisim, bolge ve kullanılan optik araclar da goz onune alindiginda bir takim parametrelerin degistirilmesine ihtiyac duyulabilir. Bu parametreler *atrack.config* dosyasi icerisinde bulunmaktadir ve 3 ayrı baslik altında kategorilenmistir: *sources* (nesnelerin sextractor ile tespiti

icin gerekli parametreler), *asteroids* (hareketli nesne olup olmadiginin testi icin gerekli parametreler), *visuals* (gorsellestirme icin gerekli parametreler). Bu parametlerin degisimine ihtiyac duyulmadigi durumlarda pipeline on tanimli parametrelerle calisacaktır. *sources* basligi altinda bulunan parametreler bir FITS goruntu uzerindeki nesnelerin dogru bir sekilde tespiti icin gereklidir ve tamamen Sextractor ile alakalidir. Biz bu parametreleri (Bertin and Arnouts, 1996)’in sonuk cisimler icin onerdigi degerlere gore belirledik. Bu sebeple bizim gelistirdigimiz pipeline’da hareketli cisimlerin tespitini etkileyen onemli parametlerin degisimi ile tespit edilen hareketli nesnelerin sayisini gosterir bir takım tablolar hazirlanmistir gsteren tablolar verilmitir . Ayrica su kesinlikle unutulmamalidir ki, tespit edilen nesne sayisi sadece bu parametlere bagli degildir. Sextractor parametreleri FITS goruntu uzerindeki tum kaynak noktaları tespiti ettigi durumda maksimum hareketli nesne sayisinin elde edilecegi unutulmamalidir. Elimizdeki *sources* parametleri bizim kullandigimiz veri seti icin en uygun parametre setidir ve baska veri setleri iin degistirilmesi gerekebilir ve bu tamamen ku llaniciya v e calisilan veri setine baglidir. Burada irdelenen parametrelerin degisimiyle, hareketli nesnelerin ardisik noktalarının dogru tanima algoritmasına gore tespit edilip edilemeyecegi ve bu parametrelerle tespit edilen hareketli nesne sayisi arasindaki iliski ortaya konulmustur. Kullanici bu sayede kendi veri setinde hangi deger araliginda parametre degeri belirlemesi konusunda bir fikri olusturacagini dusunmekteyiz. Bu parametrelerin ($TRAVEL_{MIN}$, $HEIGHT_{MAX}$, τ , SNR) algoritma icerisinde neleri kontrol ettigi ise Sec. 2’de ayrintili bahsedilmistir.

$TRAVEL_{MIN}(\delta x)$ parametresi Sec. 2.6’de ve Eq. 1’da ayrintili olarak aciklandigi gibi secilen iki noktanin HNAK’a alinma sartini belirtir. Yani hareketli nesne olmanin ilk sartidir. Secilen 3 noktadan herhangi ikisi arasindaki mesafe δx (px) degerinden az ise o nesne hareketli olarak kabul edilmez ve HNAK’a alinmaz. Tablo 1’de 2000CA30 ve Astronomia adli iki veri setine pipeline uygulanarak δx ’in degisimiyle tespit edilen hareketli nesne sayisi incelenmistir. Yapilan testlerde arsimizde hareketli nesne oldugu dogrulanmis en yavas nesnenin iki nokta arasindaki uzakligi 0.3 px olarak belirlenmistir. Fakat farkli veri setlerine

bu kadar kucuk bir δx parametresinin uygulanmasiyla *Slow Detection* sayisinda artma meydana gelecektir. Cunku bu degerin azalmasi hizalamanin mukemmel olamayacagi gercegi ve atmosferik seeingden kaynakli nesnelerin merkez kaymalarindan dolayi HNAK'a cok fazla nesne atilmasina sebep olacak ve olusturulacak dogru icin secilen uc noktalarin kombinasyonlari cok gereinden fazla artacaktır. Bu durum da yuksek CPU islem gucu ve zamani gerektirdiginden islemi uzatacak ayni zamanda da birbirine cok yakin nesnelerin ayni dogru uzerinde kabul edilmesini saglayacaktır. Yapilan testler ile Tablo 1'de de goruldugu uzere bu deger arttikca *Slow Detection* sayisi artmis ve *True Detection* sayisi azalmistir. Bu calisma sonucunda en optimum yavas hareket eden nesnelerin de tespiti acisindan 0.5 px degeri uygun bir on tanimli parametre olarak secilmistir. Su durum kesinlikle unutumamalidir ki δx ne kadar kuculurse *Slow Detection* (yavas hareketli nesnelerle birlikte cogunlukla False detectionlar) sayisi artacaktır.

2000CA30						Astronomia					
δx (px)	Real	Fast True Detection	Fast False Detection	Slow Detection	Elapsed Time (HH:MM:SS)	δx (px)	Real	Fast True Detection	Fast False Detection	Slow Detection	Elapsed Time (HH:MM:SS)
0.1	4	7	430	449	00:20:34	0.1	3	3	3	38	00:00:02
0.2	4	7	2	45	00:02:55	0.2	3	3	2	4	00:00:24
0.3	4	7	0	8	00:01:27	0.3	3	2	0	0	00:00:23
0.4	4	7	0	4	00:01:04	0.4	3	2	0	0	00:00:23
0.5	4	7	0	2	00:00:59	0.5	3	2	0	0	00:00:23
0.6	4	7	0	2	00:00:57	0.6	3	2	0	0	00:00:23
0.7	4	7	0	0	00:00:51	0.7	3	2	0	0	00:00:23
0.8	4	7	0	0	00:00:51	0.8	3	2	0	0	00:00:23
0.9	4	7	0	0	00:00:53	0.9	3	2	0	0	00:00:23
1.00	4	7	0	0	00:00:52	1.00	3	2	0	0	00:00:23
1.10	4	7	0	0	00:00:51	1.10	3	2	0	0	00:00:23
1.20	4	7	0	0	00:00:53	1.20	3	2	0	0	00:00:23
1.30	4	7	0	0	00:00:53	1.30	3	1	0	0	00:00:23
1.40	4	7	0	0	00:00:51	1.40	3	1	0	0	00:00:23
1.50	4	7	0	0	00:00:50	1.50	3	1	0	0	00:00:24
1.60	4	7	0	0	00:00:53	1.60	3	1	0	0	00:00:24
1.70	4	7	0	0	00:00:51	1.70	3	0	0	0	00:00:11
1.80	4	7	0	0	00:00:53	1.80	3	0	0	0	00:00:11
1.90	4	7	0	0	00:00:53	1.90	3	0	0	0	00:00:11
2.00	4	5	0	0	00:00:53	2.00	3	0	0	0	00:00:11
2.10	4	5	0	0	00:00:53	2.10	3	0	0	0	00:00:11
2.20	4	5	0	0	00:00:57	2.20	3	0	0	0	00:00:11
2.30	4	5	0	0	00:00:51	2.30	3	0	0	0	00:00:11
2.40	4	4	0	0	00:00:52	2.40	3	0	0	0	00:00:11
2.50	4	4	0	0	00:00:51	2.50	3	0	0	0	00:00:11
2.60	4	3	0	0	00:00:52	2.60	3	0	0	0	00:00:11
2.70	4	3	0	0	00:00:52	2.70	3	0	0	0	00:00:11
2.80	4	2	0	0	00:00:50	2.80	3	0	0	0	00:00:11
2.90	4	2	0	0	00:00:51	2.90	3	0	0	0	00:00:11
3.00	4	2	0	0	00:00:51	3.00	3	0	0	0	00:00:11

(a) 2000CA30

(b) Astronomia

Table 1: δx ($TRAVEL_{min}$) parametresinin farklı veri setlerinde değişimi ve kıyaslanması

Tablo 2’de Fig. 3’de açıklanan h ($HEIGHT_{MAX}$) parametresinin piksel cinsinden değişimine göre tespit edilen cisim miktarı iki farklı veri seti ile karşılaştırılmıştır (2000CA30, Astronomia). Bu karşılaştırma sonucunda bizim belirlediğimiz on tanımlı parametre doğruluğu MPC veri tabanı ve göz ile teyit edilmiş hareketli nesnenin tespit edildiği 0.10 px değeri seçilmiştir. Burada dikkat edilmesi gereken ve elde edilen sonuçlar sunu göstermektedir, teoride de beklendiği gibi, h_{max} değerinin artmasıyla bulunabilecek hareketli nesne sayısının azaldığıdır. Çünkü h_{max} parametresinin artması ardışık gelen 3 noktanın aynı doğru üzerinde bulunma olasılığını azalmaktadır.

2000CA30						Astronomia					
h_{max} (px)	Real	Fast True Detection	Fast False Detection	Slow Detection	Elapsed Time (HH:MM:SS)	h_{max} (px)	Real	Fast True Detection	Fast False Detection	Slow Detection	Elapsed Time (HH:MM:SS)
0.05	4	6	0	4	00:01:11	0.05	3	2	0	0	00:00:23
0.10	4	7	0	6	00:01:09	0.10	3	2	0	0	00:00:23
0.15	4	7	1	11	00:01:08	0.15	3	2	0	0	00:00:23
0.20	4	7	1	12	00:01:12	0.20	3	2	0	0	00:00:23
0.25	4	7	2	16	00:01:12	0.25	3	2	0	0	00:00:23
0.30	4	7	2	17	00:01:23	0.30	3	2	0	2	00:00:23
0.35	4	7	2	19	00:01:20	0.35	3	2	0	2	00:00:23
0.40	4	7	3	21	00:01:24	0.40	3	2	0	2	00:00:23
0.45	4	7	4	27	00:01:23	0.45	3	2	0	3	00:00:23
0.50	4	7	4	28	00:01:23	0.50	3	2	0	3	00:00:23
0.55	4	7	4	28	00:01:08	0.55	3	2	0	3	00:00:23
0.60	4	7	4	28	00:01:10	0.60	3	3	0	3	00:00:23
0.65	4	7	4	29	00:01:07	0.65	3	3	0	3	00:00:23
0.70	4	7	6	27	00:01:09	0.70	3	3	0	3	00:00:24
0.75	4	7	6	28	00:01:09	0.75	3	3	0	3	00:00:23
0.80	4	7	6	28	00:01:13	0.80	3	3	0	3	00:00:23
0.85	4	7	7	29	00:01:13	0.85	3	3	5	18	00:00:47
0.90	4	7	8	28	00:01:10	0.90	3	3	0	3	00:00:22
0.95	4	7	9	29	00:01:08	0.95	3	3	0	3	00:00:24
1.00	4	7	9	29	00:01:08	1.00	3	3	0	3	00:00:23

(a) 2000CA30

(b) Astronomia

Table 2: *HEIGHTMAX* (h_{max}) parametresinin farkli veri setlerinde degisimi ve kiyaslanmasi

Secilen ucuncu noktanin hesaplanan degerden ne kadar bir sapma ile gercek konumunda olacagini denetleyen τ degeri Sec. 2.5’da ile ayrintili olarak aciklanmisti. Tablo 3’da pipelinenin ilgili veri setlerine uygulanmasi sonucunda en optimum sure ve hareketli sayisinin bulunmasi bakimindan $1.00px$ degerin on tanimli deger olmasi kararlaştırılmıştır (mavi satirlar secilen on tanimli degerleri gostermektedir). Bu sonuclar ise τ degerinin artmasiyla ucuncu noktanin daha genis bir cap araliginda taranacagi belirtilmektedir ve bu sebeple Tablo 3’da gorulduđu gibi *Slow Detection* miktarini artırmaktadır. Bu da tespit edilen nesnelerin hareketli nesne olup olmadigi konusunda net bir sey soyleneceğini zorlastırmaktadır. Bu degerin kucuk tutulmasi ucuncu noktanin hesaplanan bolgede olacagini belirteceginden islem hem cok hizli hem de guvenilir sonuclar verecektir. Daha ayrntili bilgi icin Sec. 2.6 ve Fig. 4 incelenebilir.

2000CA30						Astronomia					
τ (px)	Real	Fast True Detection	Fast False Detection	Slow Detection	Elapsed Time (HH:MM:SS)	τ (px)	Real	Fast True Detection	Fast False Detection	Slow Detection	Elapsed Time (HH:MM:SS)
0.25	4	7	0	2	00:01:25	0.25	4	7	0	2	00:01:25
0.50	4	7	0	4	00:01:22	0.50	4	7	0	4	00:01:22
0.75	4	7	0	6	00:01:23	0.75	4	7	0	6	00:01:23
1.00	4	7	0	6	00:01:23	1.00	4	7	0	6	00:01:23
1.25	4	7	1	8	00:01:23	1.25	4	7	1	8	00:01:23
1.50	4	7	1	9	00:01:21	1.50	4	7	1	9	00:01:21
1.75	4	7	1	11	00:01:21	1.75	4	7	1	11	00:01:21
2.00	4	7	2	11	00:01:20	2.00	4	7	2	11	00:01:20
2.25	4	7	2	13	00:01:18	2.25	4	7	2	13	00:01:18
2.50	4	7	2	13	00:01:20	2.50	4	7	2	13	00:01:20
2.75	4	7	2	15	00:01:10	2.75	4	7	2	15	00:01:10
3.00	4	7	2	14	00:01:13	3.00	4	7	2	14	00:01:13
3.25	4	7	3	14	00:01:13	3.25	4	7	3	14	00:01:13
3.50	4	7	4	12	00:01:10	3.50	4	7	4	12	00:01:10
3.75	4	7	4	13	00:01:21	3.75	4	7	4	13	00:01:21
4.00	4	7	4	14	00:01:21	4.00	4	7	4	14	00:01:21

(a) 2000CA30

(b) Astronomia

Table 3: τ parametresinin farklı veri setlerinde değişimi ve kıyaslanması

Haraketli nesnelerin tespiti esnasında, SExtractor'un çok sonuk cisim olarak tespit etmiş olduğu fakat gerçekte kötü görü kalitesi ve arkaplan dalgalanmaları sebebiyle *HNAK*'a düşmüş pek çok nesne olmaktadır. İlgilenilen nesnelerin gerçekten "nesne" olup olmadığının tespiti açısından *SNR* (Signal-to-noise ratio, S/R) değeri son derece önemlidir. Bu sebeple *SNR* parametresi belirlenen değerden küçük ise o nesne *HNAK*'a alınmaz ve bu sayede algoritmanın uygulanacağı ilgili uçlu nokta kombinasyonları azalır, uzun bir CPU işleminden kaçınılarak zaman tasarrufu sağlanır. İlgili verilere pipeline çeşitli *SNR* değerlerinde uygulanarak, tespit edilen hareketli nesneler Tablo 4'de verilmıştır. Yapılan incelemeler sonucunda *SNR* on tanımlı değerinin, hem sonuk cisimlerin kaçırılmaması hem de CPU işlem yoğunluğundan en optimum parametre olması sebebiyle 10 değerinin olması tercih edilmiştir. Tablo 4'de de görüldüğü üzere *SNR* değerinin artmasıyla tespit edilen hareketli nesne miktarı azalmaktadır. İlgilenilen cismin parlak olması durumunda bu değer artırılmasını tavsiye etmekteyiz. Aksi halde bu çalışma ile belirlenen on tanımlı değer sonuk cisim taramaları için optimum değer olacaktır. Daha sonuk cisimler aranacaksa işlem zamanının artacağı göz önünde bulundurularak, bu değer düşürülebilir.

2000CA30					
<i>SNR</i>	Real	Fast True Detection	Fast False Detection	Slow Detection	Elapsed Time (HH:MM:SS)
2.5	4	7	0	6	00:01:23
5	4	7	0	6	00:01:09
7.5	4	7	0	6	00:01:23
10	4	7	0	6	00:01:08
12.5	4	6	0	6	00:01:22
15	4	5	0	6	00:01:04
17.5	4	5	0	6	00:01:09
20	4	4	0	6	00:00:56
22.5	4	3	0	6	00:00:49
25	4	3	0	5	00:00:49
27.5	4	3	0	5	00:00:46
30	4	2	0	6	00:00:46
32.5	4	2	0	6	00:00:42
35	4	2	0	6	00:00:43
37.5	4	2	0	5	00:00:40
40	4	2	0	5	00:00:42

(a) 2000CA30

Astronomia					
<i>SNR</i>	Real	Fast True Detection	Fast False Detection	Slow Detection	Elapsed Time (HH:MM:SS)
2.5	3	2	0	0	00:00:23
5	3	2	0	0	00:00:23
7.5	3	2	0	0	00:00:23
10	3	2	0	0	00:00:23
12.5	3	2	0	0	00:00:23
15	3	2	0	0	00:00:23
17.5	3	2	0	0	00:00:22
20	3	2	0	0	00:00:22
22.5	3	2	0	0	00:00:23
25	3	2	0	0	00:00:23
27.5	3	2	0	0	00:00:23
30	3	2	0	0	00:00:29
32.5	3	2	0	0	00:00:26
35	3	2	0	0	00:00:27
37.5	3	2	0	0	00:00:26
40	3	2	0	0	00:00:26

(b) Astronomia

Table 4: *SNR* parametresinin farkli veri setlerinde degisimi ve kiyaslanmasi

Tum bu sonuclar gosteriyor ki, her veri seti ve her ilgilenilen cisim farkli on tanimli parametrelere ihtiyac duymaktadir. Burada belirlenen parametreler veri setleri uzerinde iyi calisilmis hareketli nesnelerin bulunmasi icin idealdir tabii Sextractor parametrelerinin de degistirilmesi gerektiği unutulmamalidir. Bu pipeline ile hareketli nesneler hizli bir sekilde belirlenebilir ve astrometrisi ve hatta fotometrisi dahi hizli bir sekilde yapılabilir. Bu da piyasada bulunan kapali kaynak kodlu yazilimların yani sira kullanıcıların hizli, ucretsiz ve de geri bildirimler sayesinde gittikce guvenilir olarak gelisen bir yazilimi kullanmasini saglayacağı icin son derece onemlidir. A-Track elbette bu an beta asamasındadır fakat bu alanla ilgilenen gokbilimcilerin kullanmasıyla kararlı hale donuseceği kanisındayız.

Acknowledgements

Bu proje 114F477 numarası ile TUBITAK tarafından desteklenmektedir. Bu projeye verdikleri destekten dolayı basta TBTAk’a, Akdeniz niversitesi Fen Fakltesi Uzay Bilimleri ve Teknolojileri Bolumu’ne ve ayrıca projeye destek veren tum bilim adamları ve beta test kullanıcılarına tesekkür ederiz. Also, we

thank to TUBITAK for a partial support in using T100 telescope with project number ;TUG proje no;.

Gibi birsey kullanmamiz lazim. Tabii bu projede kullanilip kapatilmaya ihtiyac duyuluyorsa.

References

- Bertin E, Arnouts S. Sextractor: Software for source extraction. *Astronomy and Astrophysics Supplement Series* 1996;117:393–404. URL: <http://adsabs.harvard.edu/abs/1996A%26AS...117..393B>.
- Chen TC, Chung KL. A new randomized algorithm for detecting lines. *Real-Time Imaging* 2001;7(6):473–81. doi:10.1006/rtim.2001.0233.
- Duda RO, Hart PE. Use of the hough transformation to detect lines and curves in pictures. *Commun ACM* 1972;15(1):11–5. URL: <http://dl.acm.org/citation.cfm?doid=361237.361242>. doi:10.1145/361237.361242.
- Hough PV. General purpose visual input for a computer. *Ann N Y Acad Sci* 1962;99:323–34. URL: <http://www.ncbi.nlm.nih.gov/pubmed/14449281><http://onlinelibrary.wiley.com/doi/10.1111/j.1749-6632.1962.tb45317.x/abstract>; hOUGH, P V eng Not Available 1962/06/29 *Ann N Y Acad Sci*. 1962 Jun 29;99:323-34.
- Lang D, Hogg DW, Mierle K, Blanton M, Roweis S. Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal* 2010;139(5):1782–800. doi:10.1088/0004-6256/139/5/1782.
- Raab H. Astrometrica: Astrometric data reduction of CCD images. *Astrophysics Source Code Library*, 2012. URL: <http://adsabs.harvard.edu/abs/2012ascl.soft03012R>.
- Savanevich VE, Kozhukhov AM, Bryukhovetskiy AB, Vlasenko VP, Dikov EN, Ivashchenko YN, Elenin LV. Program of automated asteroids detection colitec

— new features and results of implementation. 2012. URL: <http://adsabs.harvard.edu/abs/2012LPI....43.1049S>.

Tewes M. a python package to quickly, automatically, and robustly identify geometrical transforms between optical astronomical images. 2015. URL: <http://obswww.unige.ch/~tewes/alipy/>.