

**T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**AYNI BÖLGEYE AİT ARDIŞIK GÖKYÜZÜ GÖRÜNTÜLERİNDEKİ
HAREKETLİ CİSİMLERİN DOĞRU TANIMA YÖNTEMİ İLE OTOMATİK
TESPİTİ**

Nurdan KARAPINAR

**YÜKSEK LİSANS TEZİ
UZAY BİLİMLERİ VE TEKNOLOJİLERİ ANABİLİM DALI**

2017

**T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**AYNI BÖLGEYE AİT ARDIŞIK GÖKYÜZÜ GÖRÜNTÜLERİNDEKİ
HAREKETLİ CİSİMLERİN DOĞRU TANIMA YÖNTEMİ İLE OTOMATİK
TESPİTİ**

Nurdan KARAPINAR

**YÜKSEK LİSANS TEZİ
UZAY BİLİMLERİ VE TEKNOLOJİLERİ ANABİLİM DALI**

(Bu tez TÜBİTAK tarafından 114F477 nolu proje ile desteklenmiştir.)

2017

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AYNI BÖLGEYE AİT ARDIŞIK GÖKYÜZÜ GÖRÜNTÜLERİNDEKİ
HAREKETLİ CİSİMLERİN
DOĞRU TANIMA YÖNTEMİ İLE OTOMATİK TESPİTİ

Nurdan KARAPINAR

YÜKSEK LİSANS TEZİ
UZAY BİLİMLERİ VE TEKNOLOJİLERİ ANABİLİM DALI

Bu tez 26/7/2017 tarihinde aşağıdaki jüri tarafından Oybirliği/Oyçokluğu ile kabul edilmiştir.

Yrd. Doç. Dr. Murat KAPLAN

Prof. Dr. Volkan BAKIŞ

Doç. Dr. Hasan Hüseyin ESENOĞLU



ÖZET

AYNI BÖLGEYE AİT ARDIŞIK GÖKYÜZÜ GÖRÜNTÜLERİNDEKİ HAREKETLİ CİSİMLERİN DOĞRU TANIMA YÖNTEMİ İLE OTOMATİK TESPİTİ

Nurdan KARAPINAR

Yüksek Lisans Tezi, Uzay Bilimleri Ve Teknolojileri Anabilim Dalı

Danışman: Yrd. Doç. Dr. Murat KAPLAN

Temmuz 2017, 49 sayfa

Bu tez çalışmasının amacı ardışık teleskop görüntülerindeki hareketli cisimleri Doğru Tanıma Yöntemi'ni (DTY) (Chen T. ve Chung K., 2001) kullanarak otomatik olarak tespit eden bir bilgisayar programı geliştirmektir. Doğru Tanıma Yöntemi, aslında, dijital görüntülerdeki doğruları tespit etmek için geliştirilmiş matematiksel bir yöntemdir. Bu yöntem, bu tez ile birlikte ilk kez teleskop görüntülerine uyarlanacak ve bu görüntülerdeki hareketli cisimlerin (asteroid, kuyruklu yıldız, vb.) tespiti için kullanılacaktır.

Güneş sistemi cisimleri üzerine yapılan gözlemsel çalışmalar sırasında gece boyunca çok sayıda teleskop görüntüsü alınmaktadır. Bu görüntüler içerisine gözlenen cisimler dışında bilinen ya da bilinmeyen hareketli cisimler de girebilmektedir. Bu görüntülerin hızlı bir şekilde taranıp işlenerek görüntülerdeki hareketli cisimlerin tespit edilmesi hem daha önce gözlenmiş cisimlerin yörüngelerinin iyileştirilmesi hem de yeni cisimlerin keşfi açısından önem taşımaktadır.

Hareketli güneş sistemi cisimlerinin çok sayıda teleskop görüntüsü kullanılarak klasik yöntemlerle tespit edilmesi yüksek zaman tüketimi ve iş gücü gerektirmektedir. Geliştirilen bilgisayar programı sayesinde ardışık teleskop görüntülerinin hızlı bir şekilde işlenmesi ve içerisindeki hareketli cisimlerin tespit edilmesi mümkün olmaktadır. Program, teleskop görüntülerinin analizinde daha önce hiç uygulanmamış olan Doğru Tanıma Yöntemi'nden esinlenerek Çoklu Görüntülerden Doğru Tanıma Yöntemi (Multiple Image Line Detection(MILD)) (Atay ve ark.,2016) kullanılarak, çalışması esnasında insan etkileşimini en aza indirecek şekilde hazırlanmıştır.

Programda kullanılacak yöntemler işlem sırasına göre şu şekilde özetlenebilir: İsteğe bağlı olarak görüntüler ön indirgemeye tabi tutulmaktadır. Görüntülerdeki, teleskobun hareketinden kaynaklanan kaymalar ve gözlem koşullarından kaynaklanan ışık şiddeti farkları düzeltilir. Hareketsiz cisimlerin (yıldızların) büyük çoğunluğu görüntülerden çıkarılarak, Ay veya yıldız kaynaklı parlama etkileri büyük ölçüde azaltılmaktadır. Görüntülerde geriye kalan ışık kaynaklarının (potansiyel hareketli cisimler) koordinatları bu kaynakların parlaklık değerleri üzerinden tanımlanan “parlaklık merkezleri” hesaplanarak bulunmaktadır. Bu koordinatlar arasından gerçekten hareketli cisimlere ait olanlar MILD kullanılarak tespit edilir. Tüm bu işlemler otomatik bir şekilde gerçekleştirilmektedir.

MILD'nin kullanıldığı bu bilgisayar programı ile gecelik gözlemlerden elde edilen teleskop görüntülerindeki hareketli cisimlerin koordinatları hızlı ve otomatik bir şekilde elde edilebileceğinden, program yeni cisimlerin keşfi için önemli bir araç vazifesi görecektir. Arşiv görüntülerinden klasik yöntemlerle hareketli cisim tespit etmek hem zaman hem de iş gücü açısından yüksek maliyetlidir. Bu program ile istendiğinde bu tür arşiv taramalarının hızlı ve otomatik bir şekilde gerçekleştirilmesi mümkün olacaktır.

ANAHTAR KELİMELEER: Asteroid, Hareketli Cisim Tespiti, Doğru Tanıma Yöntemi, Görüntü İşleme

JÜRİ: Yrd. Doç. Dr. Murat KAPLAN (Danışman)

Prof. Dr. Volkan BAKIŞ

Doç. Dr. Hasan Hüseyin ESENOĞLU

ABSTRACT

AUTOMATIC DETECTION OF MOVING OBJECT FROM SEQUENTIAL TELESCOPE IMAGES USING LINE DETECTION ALGORITHM

Nurdan KARAPINAR

MSc Thesis in Space Sciences and Technologist
Supervisor: Assistant Prof. Dr. Murat KAPLAN
July 2017, 49 pages

The aim of this thesis is to develop a computer program that uses the Line Detection Method (Chen T. and Chung K., 2001) to detect any moving object in a given series of sequential telescope images. The Line Detection Method is actually a mathematical technique developed to detect lines in digital images. This technique is applied to telescope images for the first time and used to detect the moving objects (asteroids, comets, etc.) in these images.

Many telescope images are acquired during the night long observations of various objects in our solar system. In addition to those already being observed, these images may contain other known or unknown moving objects. To discover new objects and to refine the orbital parameters of previously identified objects, it is important that these images are processed with high speed to detect all moving objects in them.

The detection of moving solar system objects from multi-frame telescope images can be time-consuming and require too much work power if one resorts to classical methods. With this proposed computer program, it will be possible to process any number of telescope images with high speed and identify the moving objects in them. The improved computer program allows you to quickly and easily move telescope images quickly. Multiple Image Line Detection (MILD) (Atay et al., 2016) has been prepared, which has never been applied to the analysis of telescope images before, will be utilized for detection and human interaction will be minimized during operation.

The methods that are used in various stages of the program can be summarized in working order as follows: The images are subject to an optional pre-reduction step. The translational shifts due to tracking errors and the intensity differences due to atmospheric changes are corrected for each image. Most of the stationary objects (stars) and Moon or star related glowing effects will be eliminated from the images. The center coordinates of the remaining light sources (moving object candidates) are calculated using a center-of-light function determined by the intensity values of the pixels in the images. The coordinates of the real moving objects among all candidates are identified using MILD algorithm. All these steps are performed automatically.

Since with this computer program using the MILD, the coordinates of the moving objects in the telescope images will be calculated automatically with high speed, the proposed program is expected to serve as an important tool to discover new objects. Detection of moving objects from archive images using classical methods is not

cost-effective in terms of time and work power. With this program, we're expecting that such archive scans will become fast and automated.

KEYWORDS: Asteroid, Moving Object Detection, Line Detection Method, Image Processing

COMMITTEE: Asst. Prof. Dr. Murat KAPLAN (Supervisor)

Prof. Dr. Volkan BAKIŞ

Assoc. Prof. Dr. Hasan Hüseyin ESENOĞLU

ÖNSÖZ

Bu çalışmada amaç aynı bölgeye ait ardışık teleskop görüntülerini daha önce bu alanda kullanılmamış bir algoritma kullanarak analiz edebilmektir. Python 3x programlama dili kullanılarak geliştirilen yöntemin yazılıma dönüştürülmesi ve uygun bir arayüz ile son kullanıcıya sunulması hedeflenmiştir. Çalışmanın özgünlüğü görüntülerin daha önce kullanılmamış bir yöntemle analizi ile hareketli güneş sistemi cisimlerinin saptanabilmesidir. Bu amaçla tek bir görüntü üzerindeki doğruları tespit eden klasik Doğru Tanıma Algoritmasından esinlenerek ardışık teleskop görüntüleri üzerinde çalışan Multiple Image Line Detection (MILD) (Çoklu Görüntüden Doğru Tanıma Algoritması) algoritması geliştirildi. Tezin içeriğinde aşamalı olarak bu geliştirme işlemi ve gerekli bileşenleri anlatıldı.

Tez çalışmam boyunca her türlü yardım ve fedakârlığı sağlayan, bilgi aktarımında sebatı ve ayrıcalıksızlığı ile örnek olan, bana bu tez çalışmasını vererek kendimi geliştirmeye yönelik de birkaç adım ileride olmamı sağlayan, çalışmamın yöneticisi ve bilimsel danışmanım Yrd. Doç. Dr. Tolga ATAY hocama teşekkürlerimi sunuyorum.

Çalışmam sırasında yaşadığım olumsuz değişimlerde desteğini gördüğüm, tez çalışmamda katkısı olan danışmanım Yrd. Doç. Dr. Murat KAPLAN'a teşekkürlerimi sunuyorum.

Hayat boyu verdikleri maddi manevi destekleri için sevgili aileme, çalışma boyunca emek ve desteklerini esirgemeyen dostlarımdan canım dostum Saliha EREN'e, değerli arkadaşlarım Gürkan ASLAN ve Nagihan ASLAN'a, oda arkadaşım Selen OY'a teşekkürlerimi sunuyorum. TÜBİTAK-114F477 kodlu projemizde emeği geçen Yücel KILIÇ'a ve TÜBİTAK-14BT100-648 kodlu proje kapsamında T100 teleskobu ile gözlem yapan arkadaşlarıma da teşekkür ederim.

İÇİNDEKİLER

ÖZET	i
ABSTRACT	iii
ÖNSÖZ	v
İÇİNDEKİLER	vi
SİMGELER VE KISALTMALAR DİZİNİ	viii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	x
1. GİRİŞ	1
2. KURAMSAL BİLGİLER VE KAYNAK TARAMALARI	4
2.1. Görüntülerin Özellikleri	4
2.1.1. Hareketli cisim için gözlem gereksinimleri	4
2.1.2. Gözlem için gerekli araçlar ve özellikleri	4
2.1.3. CCD özellikleri	5
2.1.4. FITS dosya formatı	6
2.2. Literatürdeki Hareketli Nesne Tespit Yöntemleri	6
2.2.1. Astrometrica programı	7
2.2.2. CoLiTeC programı	8
2.2.3. Asteroid Data Hunter programı	8
2.3. Çalışmada Kullanılan Bilgisayar Yazılımları	9
2.3.1. SExtractor yazılımı ve kullanım amacı	9
2.3.2. Alipy yazılımı ve kullanım amacı	12
2.3.3. f2n paketi ve kullanım amacı	12
2.3.4. PySex paketi ve kullanım amacı	13
2.3.5. Pickle ve Multiprocessing paketi ve kullanım amaçları	13
2.4 GÖRÜNTÜLERE UYGULANAN İŞLEMLER	14
2.4.1. Görüntülerde Işık Kaynağı Tespiti	14
2.4.1.1. Gauss nokta dağılım fonksiyonu	14
2.5. Görüntülerin Hizalanması	16
2.5.1. Afın dönüşümü	17
2.5.2. En küçük kareler yöntemi	19
3. MATERYAL VE METOT	21
3.1. Klasik Doğru Tanıma Algoritması	21
3.2. Çoklu Görüntülerden Doğru Tanıma Algoritması (Multiple Image Line Detection)	23
3.2.1. A-Track programı	26
4. BULGULAR VE TARTIŞMA	30
4.1. Kullanılan Görüntülerin Özellikleri	30
4.2. Benzer Programlar ile Karşılaştırılması	31
4.3. A-Track Performans Testi	33
4.3.1. HEIGHT_MAX parametresi	33
4.3.2. TRAVEL_MIN parametresi	35
4.3.3. SNR_MIN parametresi	36
4.3.4. TOLERANCE parametresi	36
4.4. Farklı Verilerle A-Track Performans Testi	37
5. SONUÇ	39

6. KAYNAKLAR	41
7. EKLER	43
Ek 1: A-Track Programının Kurulumu ve Kullanımı	43
Ek 2: A-Track Programının Arayüzü	45
ÖZGEÇMİŞ	

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

ADU Analog Digital Unit

Kısaltmalar

AD Afın dönüşümü

ADH Asteroid Data Hunter

CCD Charge Coupled Device

DTY Doğru Tanıma Yöntemi

FITS Flexible Image Transport System

GNDF Gauss Nokta Dağılım Fonksiyonu

MILD Multiple Image Line Detection

MF Matched Filter

NASA National Aeronautics and Space Administration

NEO Near Earth Object

NMF Neural Modeling Fields

PIL Python Image Library

TUG TÜBİTAK Ulusal Gözlemevi

YYA Yer'e yakın asteroidler

ŞEKİLLER DİZİNİ

Şekil 2.1. CCD'nin enerji iletim mekanizması	5
Şekil 2.2. Astrometrica programı ile veri işleme sonucunda ekran görüntüsü	7
Şekil 2.3. CoLiTeC'in geliştirdiği LookSky programından ekran görüntüsü.....	8
Şekil 2.4. Asteroid Data Hunter programının web tabanlı çalışma alanı.....	9
Şekil 2.5. Örnek görüntüde SExtractor'ın extraction aşamaları	10
Şekil 2.6. Örnek görüntüde Alipy paketinin hizalama işlemi	12
Şekil 2.7. GNDF grafiğine bir örnek.....	15
Şekil 2.8. Örnek görüntünün referans görüntü kullanılması ile hizalama (align)işleminin yapılma aşamaları	16
Şekil 2.9. Afin dönüşümünde eksenler yönünde kayma ve dönme örneği	17
Şekil 2.10. a) Pozitif doğrusal kuvvetli ilişki b) Negatif doğrusal kuvvetli ilişki.....	20
Şekil 3.1. DTY'nin doğrudaşlık için birinci şartı.....	21
Şekil 3.2. DTY'nin doğrudaşlık için ikinci şartı	22
Şekil 3.3. DTY'nin doğrudaşlık için üçüncü şartı	22
Şekil 3.4. MILD'ın doğrudaşlık için birinci şartı.....	23
Şekil 3.5. MILD'ın doğrudaşlık için ikinci şartı	24
Şekil 3.6. MILD'ın doğrudaşlık için üçüncü şartı	25
Şekil 3.7. A-Track programının çalışma şeması	28
Şekil 7.1. A-Track arayüz 1.sayfa görüntüsü	46
Şekil 7.2. A-Track arayüz 2.sayfa görüntüsü	47
Şekil 7.3. A-Track arayüz 3.sayfa görüntüsü	48
Şekil 7.4. A-Track arayüz 4.sayfa görüntüsü	49

ÇİZELGELER DİZİNİ

- Çizelge 4.1. Tablonun süre sütununda 1.veri MILD ile tasarlanılan yazılım A-Track
2.veri Astrometrica 3.veri CoLiTeC çalışma sürelerini göstermektedir.... 32
- Çizelge 4.2. Height_Max parametresi performans testi. Height_Max parametresi
[0.05-1.00] arasında 0.05 adım aralığında değişirken, diğer parametreler
Travel_min = 0.5 Tolerance = 1.0, SNR_Min= 10 olarak alınmıştır. 34
- Çizelge 4.3. Travel_Min parametresi performans testi. Travel_Min parametresi
[0.1-3.00] arasında 0.10 adım aralığında değişirken, diğer parametreler
Height_Max = 0.15, Tolerance = 1.0, SNR_Min = 10 olarak alınmıştır... 35
- Çizelge 4.4. SNR_Min parametresi performans testi. SNR_Min parametresi
[2.5-40] arasında 2.5 adım aralığında değişirken, diğer parametreler
Height_Max = 0.15, Tolerance = 1.0, Travel_Min= 0.5 olarak alınmıştır 36
- Çizelge 4.5. Tolerance parametresi performans testi. Tolerance parametresi
[0.25-4.00] arasında 0.25 adım aralığında değişirken, diğer parametreler
Height_Max = 0.15, Travel_Min = 0.5, SNR_Min= 10 olarak alınmıştır. 37
- Çizelge 4.6. Test edilen 11 veri setine ait sonuçlar 37

1. GİRİŞ

Güneş sistemi cisimleri üzerine yapılan gözlemsel çalışmalar sırasında gece boyunca çok sayıda teleskop görüntüsü alınmaktadır. Bu görüntüler içerisine gözlenen cisimler dışında bilinen ya da bilinmeyen hareketli cisimler de girebilmektedir. Literatürde nesne tespiti konusunda birçok yazılım, yöntem v.b. vardır. Bu görüntülerin hızlı bir şekilde taranıp işlenerek görüntülerdeki hareketli cisimlerin tespit edilmesi hem daha önce gözlenmiş cisimlerin yörüngelerinin iyileştirilmesi hem de yeni cisimlerin keşfi açısından önem taşımaktadır. Güneş sistemi cisimlerinin çok sayıda teleskop görüntüsü kullanılarak klasik yöntemlerle tespit edilmesi yüksek zaman tüketimi ve iş gücü gerektirmektedir. Bu işlem gücü gereksinimini azaltabilmek için geliştirilen ve uyarlanan çeşitli matematiksel yöntemler literatürde mevcuttur. Aynı şekilde işlem gücünü azaltmak için çeşitli topluluklar (Pan-Stars, Catalina v.b.) tarafından kendi teleskop sistemleri için geliştirdikleri yöntemler vardır. Ancak, bu toplulukların yazılımlarında kullandıkları yöntemleri paylaşımına kapalıdır.

Literatürde bu konuda çeşitli çalışmalar bulunmaktadır. Genel olarak baktığımızda yapılan bu çalışmalar temelde aynı konu üzerinde olsa da ara aşamalarda ve sonuçlarda tamamen farklılık göstermektedir. Literatürde farklı yöntemler ve yazılımlar vardır. Kısaca bahsetmek gerekirse bunlardan bazıları; Multi-Aperture Asteroid Detection System algoritması kullanılarak geliştirilen bir sistemde çapı yaklaşık 20 santimetre olan çok sayıda küçük teleskop ile hareketli cisim tespit edilmeye çalışılmıştır. Bu sistemin iyi çalışması; CCD'nin algılama performansına ve arka planın ne kadar parlak olduğuna bağlıdır. Bu işlemlerin tamamlanması için 40-60 arasında görüntüden oluşturulmuş birleştirilmiş yeni bir görüntü gerekmektedir (Cain vd., 2007).

Bir diğer yöntemde ise *Neural Modeling Fields (NMF)* kullanılmıştır. Öncelikle bu yöntemi anlayabilmek için yapay sinir hücrelerinden en genel biçimde bahsetmek gerekiyor. Yapay sinir hücreleri her bir girdi örüntüsüne karşılık y çıktı sinyali oluşturmakla ve daha sonra bu sinyali diğer hücelere iletmekle görevlidir. Her giriş-çıkış çifti arasındaki doğrusal ilişkinin yönünü ve şiddetini gösteren ağırlık değerleri vardır. Bu değer her yeni ikiliye göre tekrar hesaplanır ve amaca ulaşılmışsa saklanır. Bu hesaplama sürecine öğrenme denir. Durağanlık sağlandığı zaman hücre öğrenmesini tamamlamıştır (Cinsdikici, 1997). Görüntülerde kullanabilmek için bu yöntemde aranan cisme en yakın komşuluğa ulaşana kadar öğrenme yani tespit etme işlemine devam edilir. NMF yöntemini asteroid tespit etmede kullanabilmek için çok sayıda gözlem verisi bilinen asteroide en yakın komşuluğa ulaşana kadar görüntüler kümelenerek bölünür. En son bölgede asteroid tespit edilmiş olacaktır. Bu yöntem için de görüntü sayısı arttıkça işlem süresi artar (O'Connor vd., 2012).

Bu alanda kullanılmış olan farklı bir yöntem olan “Kd-Tree”, “K-dimensional tree” açılımında “K boyutlu ağaç” anlamı taşımaktadır. Bu yöntemde uzay, x-y eksenleri ile paralel olan dikdörtgen prizması şekilde tasarlanıp, belirli bir maksimum üçgen sayısına kadar alt uzaylara bölünür. Bu bölünme şekli de aynı yapıdadır ve daha sonraki aşamalarda aynı özelliğin korunması modelin korunmasını sağlayacaktır. Kd-Tree’de uzay eşit alanlara bölünmek zorunda değildir. Bu yöntemde bölünen alt uzayların ağırlıklarının, belli bir aralığa göre yapılan arama olduğu için, eşit olması beklenir. Bu yöntemin asteroid yakalaması için her gece bir bölge için 15 dakika arayla

dört görüntü alınmakta, bu şekilde bütün gökyüzü taraması yapılmaktadır. Algoritmanın verimli çalışabilmesi için bu işlemleri 7-14 gece tekrarlamak gerekmektedir. Algoritmanın yüksek hesaplama gereksinimleri nedeniyle, alınan görüntüleri işleyebilmek için bir küme hesaplama aracı kullanılmalıdır. Bu yöntem PAN-STARRS grubu tarafından kullanılmaktadır (Kubica vd., 2007). Yazılım ve algoritmaların benzer şartlarda elde edilmemiş görüntüler için uygun olmaması sebebiyle halka açık bir yazılımları yoktur.

Bu yöntemlerin dışında hareketli cisim tespiti için kullanılan yazılımlar da vardır. Bu yazılımlardan ilki Mohanty (1981) tarafından ilk geliştirildiğinde yıldız ardaşlarında uyduların konumlarını tespit etmek için kullanılmış olan *Matched Filter Processing (MF)* yöntem ile geliştirilen SALTAD'tır. Temel yöntem iki aşamadan oluşur. Bir, ardaşları baskılayarak sinyal etkisini ortadan kaldırmak, iki, sinyal oluşturabilecek bileşeni artırmak. YYA'ları yakalayabilmek için geliştirilmiş bir yazılımdır. MF yöntemi asteroidin konumunun, hız ve yönünün bilinmesi ile çalışabilmektedir. Gözlem verileri arasında bir saatten daha az zaman farkı kısıtlaması vardır. Görüntüler arasında zaman kayması sorunu oluşmaktadır (Gural vd., 2005).

Bir diğer yazılım örneği ise *Automatic Detection Algorithm*'de küçük hareketli cisimleri tespit etmek için geliştirilmiştir. Bu yazılımda asteroidin konumunun bilinmesi gerekmektedir ve yöntemde hareketli nesnelerin olduğu kısım kesilerek alt-görüntüler haline getirilir. Bu alt-görüntüler ana görüntüden çıkarılarak median-görüntü elde edilir. Ancak bunun için her bölgenin yaklaşık 40 görüntüsü alınıp işlenmelidir. Grubun kapalı ve ticari hale dönüştürdükleri yazılımlarının sıradan masaüstü bir bilgisayarda 40 görüntü işlemek için ortalama 3 saat çalışması gerekmektedir (Yanagisawa vd., 2007).

Bu çalışmada esinlenilen Doğru Tanıma Yöntemi (DTY), aslında, dijital görüntülerdeki doğruları tespit etmek için geliştirilmiş matematiksel bir yöntemdir. Bu yöntem, ilk kez teleskop görüntülerine uyarlandı ve bu görüntülerdeki hareketli cisimlerin (asteroid, kuyruklu yıldız, vb.) tespiti için kullanıldı. Aslında bir doğru oluşturabilecek veya oluşturamayacak noktaların tespitinde, sayısal görüntünden elle kontrol ile piksel komşuluklarına bakarak da bu işlemi gerçekleştirebilir. Ancak sayısal görüntünün boyutuna göre piksel sayısından yola çıkılarak bu işlem için, piksel x piksel kere işlem gerekmektedir. Yani işlem süresi katlanarak artacaktır. En önemlisi de bu yöntem uyarlanmadıkça ardışık görüntülerde temel haliyle kullanılamaz.

Çalışmanın ikinci bölümünde kullanılan araçlar hakkında genel bilgi verildi, üçüncü bölümde görüntülere uygulanan ön işlemlerden bahsedildi. Üçüncü bölümde çalışmada ihtiyaç duyulan ek modüllerden bahsedildi. Dördüncü bölümde klasik DTY'den uyarlanılan Multiple Image Line Detection (MILD) (Çoklu Görüntüden Doğru Tanıma Algoritması) yöntemi kullanılarak geliştirilen A-Track programından ve bu programın kullanıcıların kolay kullanımı için tasarlanılan arayüzden bahsedildi. Tezde Tübitak Ulusal Gözlemevi'nden TÜBİTAK-14BT100-648 kodlu proje kapsamında T100 teleskobu ile alınmış asteroid gözlem verileri ve TÜBİTAK-14BT100-327 kodlu proje kapsamında T100 teleskobu ile alınmış asteroid gözlem verileri kullanılmıştır.

Toplamda yaklaşık olarak 30'u aşkın veri seti kullanılarak hem performans testi hem de yeni asteroid keşfi amaçlandı. Tezin bulgular bölümünde farklı gözlem verileri

için elde edilen sonuçlar sunuldu. Yedinci bölümde MILD'ın literatürdeki algoritmaların çalışma prensiplerine göre neden daha hızlı ve daha güvenilir sonuçlar verdiğinden detaylı olarak bahsedildi. Ekler bölümünde A-Track programını nasıl indirilip kullanılacağı ve tasarlanan arayüz hakkında bilgiler verildi.

2. KURAMSAL BİLGİLER VE KAYNAK TARAMALARI

2.1. Görüntülerin Özellikleri

2.1.1. Hareketli cisim için gözlem gereksinimleri

Asteroid gözlemlerinin yıldız gözlemlerinden farkı gözlem tekniklerinin yıldız gözlemlerinden farklı olmasıdır. Bu gözlemlerde daha yüksek hassasiyette konum bilgisine, dolayısıyla da görüntü kalitesi yüksek gözleme ihtiyaç duyması ve kısa poz sürelerine sahip gözlemler yapılmasıdır. Gelişen teknoloji sayesinde asteroid tespit etme yöntemleri zaman içerisinde değişimlere uğramıştır. İlk çalışmalar bilinen gezegen veya cisimlerin yörüngesindeki tedirginlikler ile teorik saptamalarla başlamış ancak daha sonraları gözlemlerle desteklenmiştir. Gözlemlerle yapılan hareketli cisim yakalama teleskop özellikleriyle sınırlıdır. Teleskobun bulunduğu gözlem ve hava koşulları, kullanılan CCD'nin kalitesi sönük ve hızlı olan asteroidlerin yakalanmasını etkiler.

Günümüzde hareketli cisim tespit etme yöntemleri yazılımlarla desteklenerek gözlem verileri daha ayrıntılı incelenebilmektedir. Giriş bölümünde bahsedilen teorik yöntemlerden bazıları bu teknolojik gelişime örnek olarak kullanıldıkları teleskoba ve gözlem tekniklerine bağlı olarak geliştirilmiştir.

2.1.2. Gözlem için gerekli araçlar ve özellikleri

Astronomide ideal bir dedektör;

1. Bütün dalga boylarında çalışabilen,
2. Gelen bütün fotonları sayım yapabilen,
3. Bu sayının kayıtlanıp işlenmesini sağlayan,
4. Uzun zaman kararlı algılayan,
5. Algılama yüzeyi geniş bir cihazdır.

Ancak tüm bu özelliklere sahip ideal bir dedektör yoktur. Bir dedektörün ideale yakınlığını ifade etmekte kullanılan parametreler bulunmaktadır. Bunlar;

1. *Kuantum Etkinliği*; detektöre gelen ışınım miktarından hangi oranda yanıt alınabildiğini gösterir. QE ile gösterilir ve

$$QE = \text{ölçülen foton sayısı} / \text{gelen foton sayısı}$$

2.1

denklemleriyle açıklanır.

2. *Tayfsal Yanıt*; dedektöre farklı dalga boylarında ve eş enerjili ışınlar gönderildiğinde her dalga boyunun kendine göre vereceği yanıt farklı olacaktır.

3. *Doğrusallık*; detektörden çıkan sinyal ile gelen ışık miktarıyla doğru orantılıysa bu dedektör için “doğrusaldır” denir.

4. *Dinamik Aralık*; kayıpsız ölçülebilen maksimum ve minimum sinyal seviyelerinin oranıdır. Dedektörün çok yüksek ve düşük sinyallerde aynı anda ölçebilme yetisidir. Yani detektördeki piksel kalitesini belirler.

5. *Gürültü*; çıktı sinyalinde her zaman ortam koşullarından dolayı bozulmalar olacaktır. Bu bozulmalar “Gürültü” olarak adlandırılır. Temel gürültü kaynakları: (1) Gözlenen kaynaktan gelen foton gürültüsü, (2) Gök arkalan foton gürültüsü ve (3) Aletsel gürültü. Bu belirsizlik sinyal/gürültü oranı(S/N) parametresi ile ölçülür.

6. *Ayrırma Gücü*; bir dedektörün üzerinde oluşan görüntüde iki yakın cisim ayırt edebilme yeteneğidir. Teleskop ve benzeri optik cihazlarla bütünleşik çalışan dedektörlerde hem optik cihazın hem de dedektörün ayırma gücü yeteneklerinin bileşkesi sonuç bir ayırma gücü oluşturur.

2.1.3. CCD özellikleri

CCD (charged-coupled device): şu anda en yaygın kullanılan dedektör türüdür. Elektromanyetik dalga formundaki enerji akısını ölçmemizi, aynı zamanda kaydederek incelememize olanak sağlayan araçtır. CCD’ler ışığı, silikondan yapılmış bir alanda, elektronik olarak dönüştürür. Daha sonra alınan akı sayısallaştırılıp görüntü dosyası formatında bilgisayara aktarılır. En düşük enerji bandına Değerlik Bandı, en üstteki band ise İletim Bandı olarak adlandırılır.



Şekil 2.1. CCD’nin enerji iletim mekanizması.

CCD’nin her bir noktası yani pikseller, algılanan şiddete bağlı olarak sinyal üretmektedir. Bir CCD’nin görüntü üretmesi için;

1. Fotonları foto elektronlara dönüştürme,
2. Foto elektronları piksellerde toplama,
3. Piksellerdeki elektronları saymak için yükü transfer etme,
4. Her pikseldeki elektronları analog-sayısal dönüştürücüler ile sayma,
5. Piksel değerlerini koordinatları ile birlikte kaydetme.

- Bu çalışmada görüntü alınan teleskobun SI 1100 CCD özellikleri;
- 4096(H)x 4097(V) Tam Frame CCD Görüntü Alanı (Piksel Formatı)
 - 15µm x 15µm Pixel Boyutu
 - 61.44 mm x 61.455 mm Görüntü Alanı

vardır.

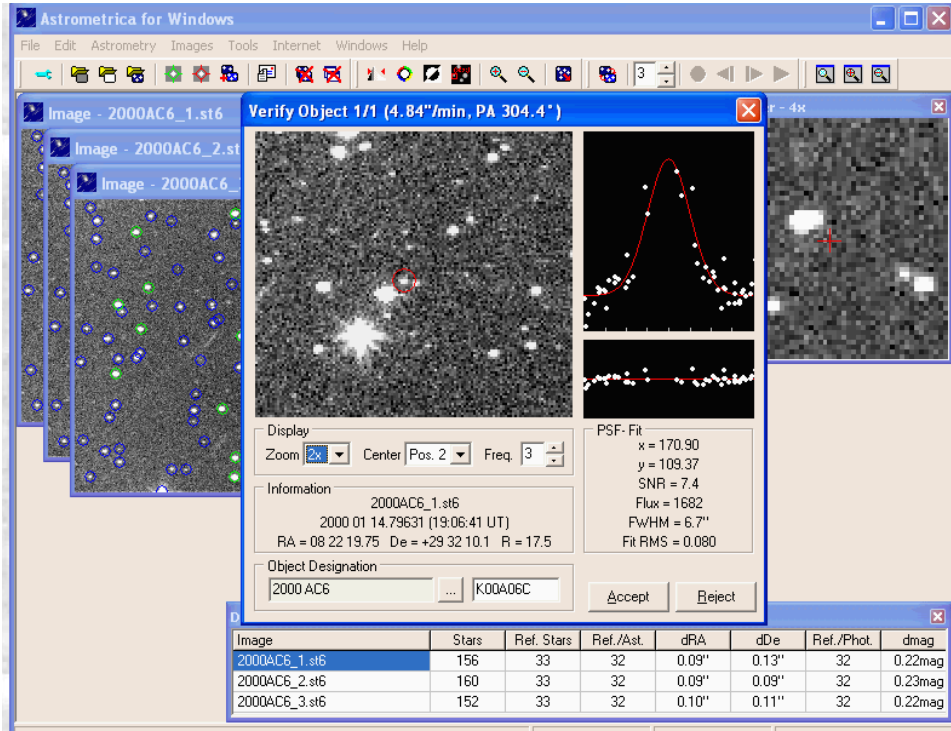
2.1.4. FITS dosya formatı

FITS dosyaları matris görüntü dosyalarıdır. Flexible Image Transport System (FITS) açılımında temel görüntü bilgilerini oluşturan bilgiye sahip genellikle astronomik amaçlı alınmış dosya formatıdır. FITS dosya formatı 1981 yılında genel bir standart oluşturulmuş ve 2008 'den beri 3.0 formatı kullanılmaktadır. FITS dosyasında saklanan meta verileri ASCII header bilgisinden okunabilir. Bir FITS dosyasında birden fazla türden dosya saklanabilir. Örneğin aynı dosyada hem x-ışın hem de kızıl öte pozlarını saklamak mümkündür. FITS dosyaları genellikle görüntü olmayan yani foton demetleri, spektrumları, veri küpleri gibi yapılandırılmış çoklu görüntü verilerini depolamak için kullanılır. C, C ++, C #, Fortran, IGOR Pro, IDL, Java, Julia, LabVIEW, Mathematica, MATLAB, Perl, PDL, Python ve R dahil olmak üzere bilimsel çalışma için kullanılan programlama dilleri için FITS destekleyen kütüphaneler vardır.

2.2. Literatürdeki Hareketli Nesne Tespit Yöntemleri

Hareketli nesne tespitinde günümüze kadar yazılmış ve kullanımı benimsenmiş üç farklı program mevcuttur. Bu programlar genel bir kitle tarafından tercih edilen kapalı kaynak kodlu ve ücretli yazılımlardır. Bölüm 2.2.1'de bahsedilen Astrometrica programının 120 günlük bir deneme sürümü vardır. Bölüm 2.2.2'deki CoLiTeC ise deneme sürümü olarak kısıtlı özelliklerinin kullanılabilirliği LookSky programı vardır.

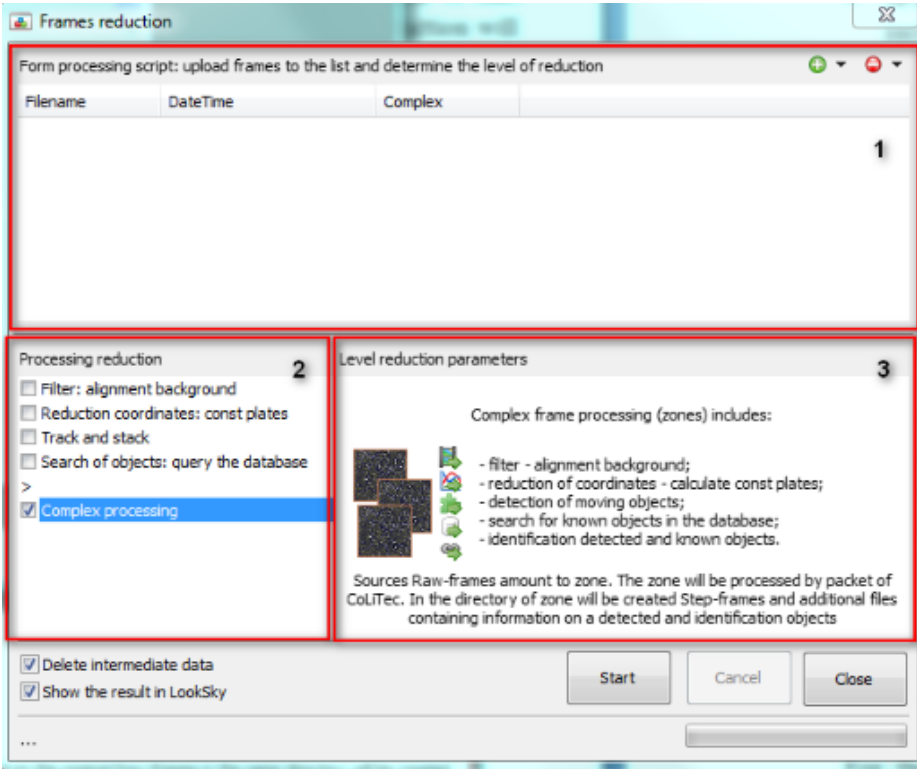
2.2.1. Astrometrica programı



Şekil 2.2. Astrometrica programı ile veri işleme sonucunda ekran görüntüsü

Astrometrica astronomlar tarafından çok tercih edilen asteroid tespit ve keşfetme aracıdır. Bu programın algoritması açık kaynak kodlu değildir ve aynı zamanda ücretli bir yazılımdır. Astrometrica kullanılırken işlemek istenilen görüntünün sağ açıklık ve dik açıklık değerleri doğru bir şekilde girilirse cisim tespit etme işlemini başarılı bir şekilde yapabilmektedir. Ancak bu değerleri tam doğru girilebilmesi için başka programlara (astrometry.net, Asplot v.b.) ihtiyaç duyulmaktadır. Bu işlem sonucunda düzgün olmayan görüntüleri hizalama işlemini kullanıcının tamamlaması gerekmekte ve bu işlem için de yine dışarıdan desteğe (Aladin) ihtiyaç duyulmaktadır. Bunların tamamlanması sonucunda tespit edilen cisimlerin tek tek doğrulanması gerekmektedir.

2.2.2. CoLiTeC programı



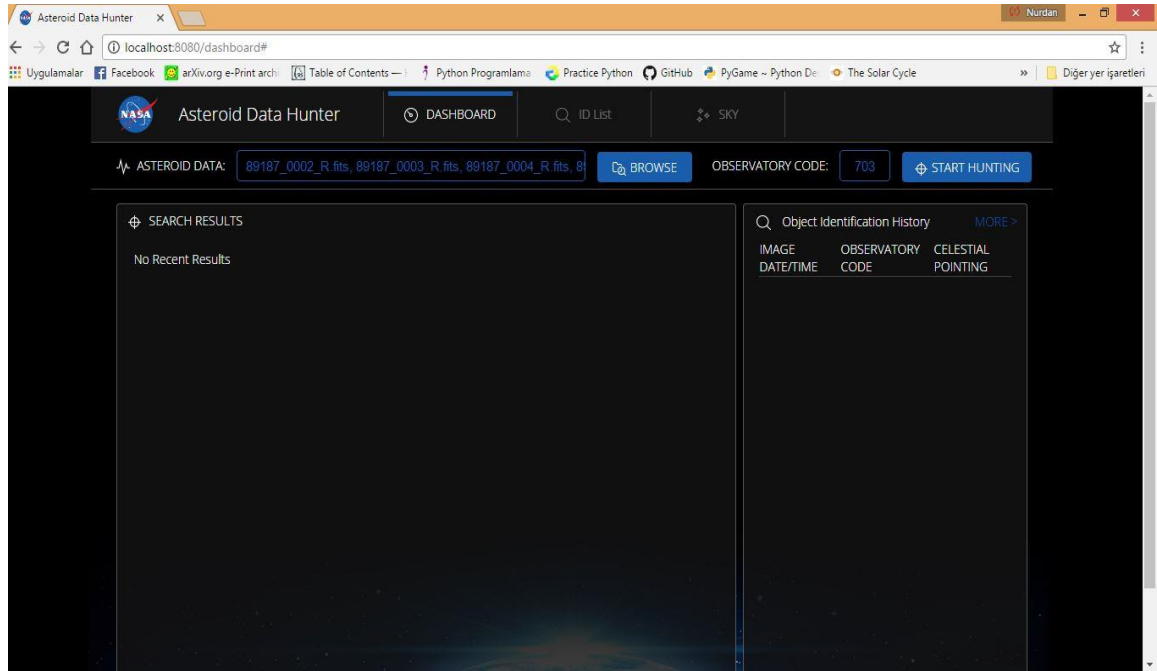
Şekil 2.3. CoLiTeC'in geliştirdiği ücretsiz LookSky programından ekran görüntüsü

CoLiTeC asteroid tespit etme konusunda tercih edilen daha yeni bir programdır. Bu programın algoritması açık değildir ve aynı zamanda ücretli bir yazılımdır. Programın tüm özelliklerini kullanabilmek için lisans ücreti ödemek gerekiyor. Bunun yanı sıra kullanıcılar için deneme yapabilecekleri LookSky isimli program kısıtlı özellikler ile kullanıma açıktır. Programı kullanmadan işlenecek görüntülerin alındığı teleskop-CCD sistemine ait temel parametrelerin programa girilmesi gerekmektedir. Bu parametreler görüntünün merkez koordinat değerleri ve teleskop sisteminin parametre değerleridir. Görüntülerde hizalama hatası veya başka bir problem çıkarsa bunu tüm aşamalar bittikten sonra kullanıcıya belirtmektedir. Bu durumda uygun olmayan görüntünün çıkartılıp işleme baştan başlanması gerekmektedir.

2.2.3. Asteroid Data Hunter programı

National Aeronautics and Space Administration (NASA) tarafından 2015 yılında geliştirilen kapalı kaynak kodlu yazılımdır. Astreroid Data Hunter (ADH) programı hareketli cisimleri yakalamak için .fits uzantılı görüntüler yerine .H uzantılı özel görüntü formatına ihtiyaç duyduğu için yeterince pratik değildir. Diğer programlardan farklı olarak web arayüzü kullanılarak çalıştırılmaktadır. FITS formatında dosyalarla çalışılabileceği NASA tarafından belirtilse de bu tez çalışmasında diğer programlarda kullanılan veriler denendiğinde çalışma tamamlanamadan arka plan hatası vermektedir.

Bu sebeple, programın mevcut sürümüyle henüz yaygın kullanıma uygun olmadığı sonucuna varılmıştır.

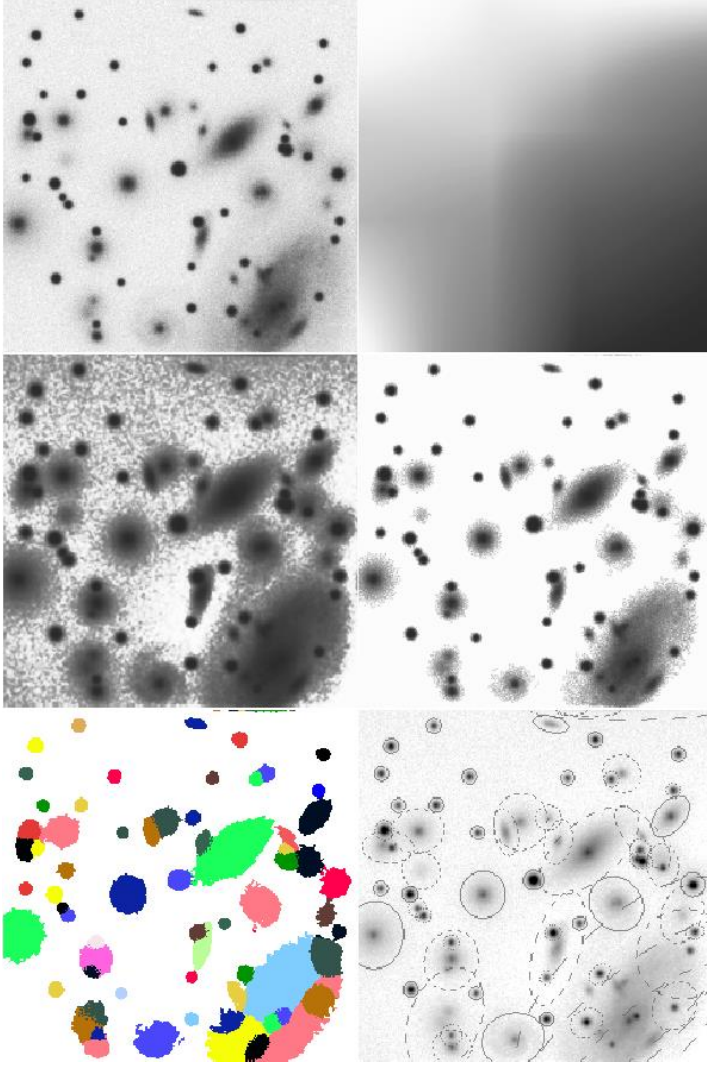


Şekil 2.4. Asteroid Data Hunter programının web tabanlı çalışma alanı

2.3. Çalışmada Kullanılan Bilgisayar Yazılımları

2.3.1. SExtractor yazılımı ve kullanım amacı

Verilerde kaynak tespiti ve kısmen hizalama aşamasında *SExtractor* programının 2.10.2 sürümü kullanılmıştır.



Şekil 2.5. Örnek görüntüde SeXtractor'ın extraction aşamaları

SExtractor yazılımı verilen bir CCD veya fotoğraf plağı (taranarak sayısallaştırılmış) görüntüsü üzerinde, belirtilen özelliklere sahip kaynak noktalarının tespitini yaparak, FITS binary tablo 2 formatında katalog dosyası oluşturur. Program sex isimli monolitik¹ yapıda tek bir çalıştırılabilir dosyadan oluşur. Çalışma sırasında yardımcı dosyalara ihtiyaç duyar. Bunlar; konfigürasyon dosyası, parametre dosyası, eğer görüntüye filtre uygulanacaksa filtre dosyası ve neural network (yapay sinir ağı) ağırlıklandırma dosyasıdır.

¹ monolitik: Tek bir parça olarak oluşturulmuş büyük kalıp şeklinde uygulamalar, yazdığımız uygulamanın veri erişim (data access), sunum (presentation), servis (service), katmanlarının hepsi tek bir uygulama içinde oluşturulmuş ve dağıtılmamışsa, bu tarz uygulamalara denir.

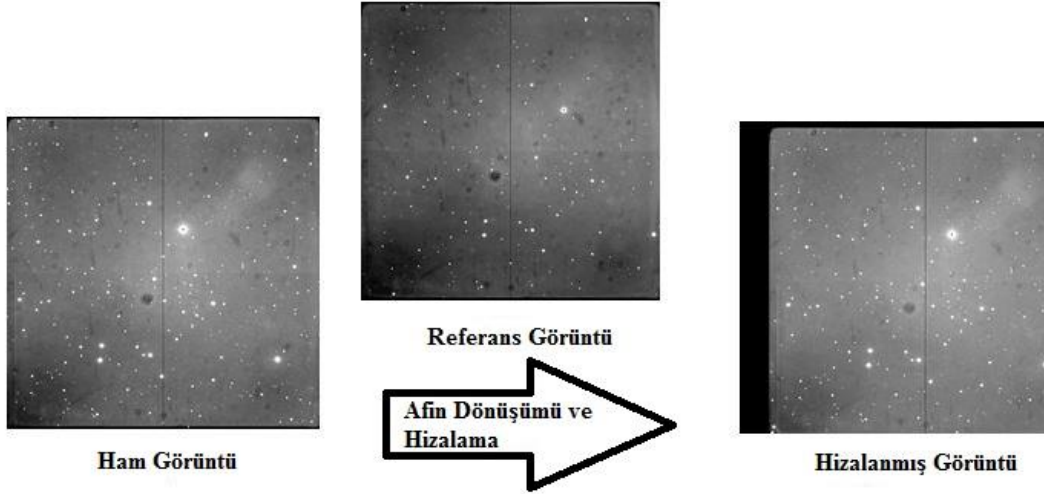
Bu çalışmada kaynak tespiti için *astrometry.net*'de denenmiş ancak daha hızlı sonuç vermesi ve kolay erişilebilirliği sebebiyle SExtractor tercih edilmiştir. Bu çalışmada öncelikle SExtractor yazılımını kullanılmasının sebebi, işlenmek üzere verilen görüntü setinden hareketsiz cisimlerin merkez koordinatlarının belirlenmesidir. Bu işlem için referans görüntüde var olan tüm cisimlerin tespitinden sonra her bir görüntü için kataloglar oluşturulur. Bu aşamada her bir görüntü için *.cat* uzantılı dosya oluşturulur. Bu dosya daha sonra tüm dosyaların toplandığı master dosyası için kullanılmak üzere saklanmaktadır. Ayrıca, incelenen aday cisimlerin belirli durumlara göre dahili işaret kodları vardır. Bunlar;

- 1, cismin komşusu varsa, aydınlık ve yeterince yakınsa,
- 2, cisim aslen bir diğer cisim ile karışmışsa,
- 4, cismin en az bir pikseli doymuşsa (ya da doymaya çok yakınsa),
- 8, cisim kesik ise (bir görüntü sınırına yakınsa),
- 16, Objektifin diyafram açıklığı verileri eksik veya bozuksa,
- 32, cismin izofotal² verileri eksik veya bozuksa,
- 64, cisim tam olarak seçilemiyorsa,
- 128, bellekten taşma extraction (çıkarma, eleme) sırasında meydana geldiğinde, dahili işaret kodu olarak kullanılır.

Bu amaçla Sextractor yazılımı ile *fits* uzantılı dosyaların header bilgilerinden yararlanarak (Sextractor kendi parametre değerlerine göre) görüntülerdeki her bir kaynağın x-y koordinat değerleri elde edilmektedir. MILD algoritmasının kullanıldığı aşamada, referans olarak verilen görüntüye göre tüm görüntülere match (birleştirme) işlemi yapılarak kaynak eşleştirmesi sonucu sabit cisimler tespit edilmiştir.

² Eşit yüzey parlaklığının noktalarını birleştiren bir galaksi, bulutsu veya diğer gökssel nesne diyagramında veya görüntüsünde bir çizgi.

2.3.2. Alipy yazılımı ve kullanım amacı



Şekil 2.6. Örnek görüntüde Alipy paketinin hizalama işlemi

Alipy astronomik görüntüler arasında geometrik dönüşümleri hızlı bir şekilde otomatik olarak tanımlamak için kullanılan bir Python paketidir. *Alipy* ile referans görüntüye göre astronomik görüntülerin geometrik çözümlerinin yapılması için tek alan yıldızları kullanılır. Her bir görüntünün piksel boyutu, yönelimi, odak düzlemi ve filtresi farklıdır. Bu hizalama işlemi için *alipy* iki farklı yöntem kullanır. Bu yöntemler aynı zamanda *scipy*'in da kullandığı afin dönüşümü ve IRAF'ın (*geomap/gerister*) metodlarıdır.

Bu modülün kullanılmasının en temel sebebi hızlı bir paket olmasıdır. Bu çalışmada *alipy*'in yöntemlerinden basit afin dönüşümü ile FITS görüntülerinin hizalaması yapıldı. Yani *alipy* paketindeki kimlik okuma ve hizalama fonksiyonlarından yararlanarak önce görüntülerin bilgileri listelenmiş ve hizalama işlemi yapıldıktan sonra görüntülerin özellikleri korunarak görüntüler şekillendirilmiştir. Daha sonra *align* modülündeki dönüşüm fonksiyonundan yararlanılarak basit afin dönüşümü görüntülere uygulanmıştır.

2.3.3. f2n paketi ve kullanım amacı

f2n FITS formatındaki dosyaları png formatına dönüştürmek için kullanılan küçük bir Python paketidir. Hızlı görselleştirme için HTML sayfalarda PNG formatındaki dosyalar kullanılır. Bu yüzden modülde 'fits to net' fonksiyonu çağrılır. Ancak *f2n* dosya dönüştürme ötesinde bazı ekstra özelliklere sahiptir. Bu özelliklerden bazıları kırpma, yeniden oluşturma, maskeleme, z ölçeklerini seçebilme, kolayca açıklama eklenebilme ve hatta tek bir PNG'den birden çok resim oluşturabilmedir. *f2n* paketi çalışırken, Python 2.5 sürümü ve üst sürümlerinde *numpy* ve *pyfits* paketlerini, Python Image Library (PIL) paketinin 1.1.6 sürümü ve üst sürümlerini kullanmaktadır.

2.3.4. PySex paketi ve kullanım amacı

Alipy paketinin alt paketi SExtractor ile Python'ı birbirine bağlar. PySex ile SExtractor katalogları oluşturularak işlenmesi sağlanır. Bu paket ile FITS görüntülerinden SExtractor kataloğu için girdi dosyalarını alır ve gerekli parametreler çağırılarak işlem yapılır. Sonuçta istediğimiz uzantıda çıktı dosyası oluşturabilir.

Her bir görüntünün SExtractor parametrelerini içeren SExtractor kataloğu oluşturulur ve bu dosyaya cat dosyası denir. Bu paket isteğe bağlı parametrelerini SExtractor'a iletir ve sonuç dosyasında olmasını istediğimiz parametreleri cat dosyası oluşturup kaydedebilmek için kullanılır.

2.3.5. Pickle ve Multiprocessing paketi ve kullanım amaçları

Python nesnelerini istenilen formata dönüştürmek ve paralel programlama özelliklerinin uygulanabilmesi için gerekli modüldür. Karmaşık veri türlerini kodlamak ve hata ayıklamayla uğraşmak yerine *Pickle* modülü ile düzenleme sağlanır. Bu modül ile saklanabilen ve farklı aşamalarda kullanılabilen Python nesneleri oluşturmanın en temel yoludur. Ayrıca, paralel programlama ile bilgisayarın işlemci gücünü tam kapasite kullanabilmek Multiprocessing modülü ile sağlanır. Bu modül Pickle modülü ile uyumludur ve birlikte kolaylıkla kullanılabilir.

Bu şekilde tek çekirdek üzerinde değil işlemcinin tüm çekirdeklerine olabildiğince eşit miktarda iş yükü dağıtılmış olacak ve yazılımın daha hızlı çalışması sağlanmış olacaktır. Yapılan testlerde hareketli nesnelerin tespiti amacıyla kullanılacak olan görüntülerin sayısı artsa bile 3 ila 15 görüntü arasında aynı performans alınmıştır. Diğer arayüzü olan programlara kıyasla daha hızlı çalışması 5. bölümde detaylı olarak incelenmiştir

2.4. Görüntüleri Uygulanan İşlemler

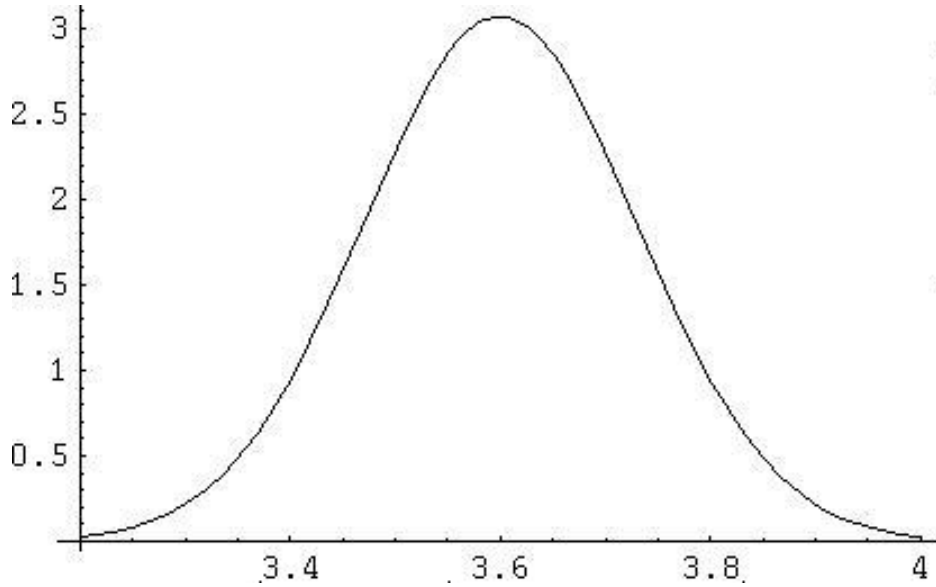
Asteroid gözlemlerinin yıldız gözlemlerinden farklı olduğuna 2.1.1. bölümde değinilmişti. Bu gözlem gereksinimlerinin oluşturduğu sorunların etkisinde kalan görüntüler belirli elemelerden geçirilmelidir. Bu çalışmada amaca yönelik bazı elemeler uygulanıldı. İlk olarak en önemli detay, analiz edilecek görüntülerin FITS formatında olması ve FITS (Anonim 2017) dosyası standartlarına uygun olması kullanıcıdan beklenmektedir. Bunun dışında geliştirilen algoritma ile bir ön indirgeme gereksinimi duyulmamaktadır. Her işlem indirgenmiş görüntüler ve ham görüntülerle test edildiğinde benzer sonuçlar alınacaktır.

2.4.1. Görüntülerde ışık kaynağı tespiti

Görüntülerde Gauss Nokta Dağılım Fonksiyonu (GNDF) algoritmanın kaynak tespiti kısmında SExtractor ile kullanıldı. Işık kaynaklarının belirlenmesi *sources* adı verilen Python modülü içerisinde bulunan **make_catalog** isimli fonksiyon tarafından yapılmaktadır. Bu fonksiyon, **alipy** modülü içerisindeki **run** isimli fonksiyonu çağırır. Bu fonksiyon görüntülerdeki noktaları tespit ederek liste halinde kullanıcıya verir ve istediği uzantıda kaydetmesini sağlar. Bu aşamada kullanılan parametreler konfigürasyon dosyasının *sources* kısmında bulunur. Analiz edilecek görüntülerdeki ışık kaynakları SExtractor programı ile belirlenerek görüntü adı ve *.pysexcat* uzantısı ile *atrack/* dizinine bir katalog dosyası olarak kaydedilir. Bu dosyalarda, görüntülerde belirlenen her bir cismin dahili işaret kodu (FLAGS), x ve y koordinatları (X_IMAGE, Y_IMAGE), akısı ve akıdaki hata (FLUX_AUTO, FLUXERR_AUTO), yerel arkaplan değeri (BACKGROUND), yarı-doruk genişliği (FWHM_IMAGE) ve uzama katsayısı (ELONGATION) bulunmaktadır (Anonim 2017).

2.4.1.1 Gauss nokta dağılım fonksiyonu

Normal dağılım olarak bilinen Gauss Nokta Dağılım Fonksiyonu (GNDF) çok farklı alanlarda uygulamalarına rastlanan bir sürekli olasılık dağılım ailesidir. Bu dağılım ailesinin her bir üyesi sadece iki parametreyle tam olarak tanımlanabilir. Bunlar *konum* gösteren ortalama (μ , aritmetik ortalama) ve *ölçek* gösteren varyanstır (σ^2 , "yayılım"). Standart nokta dağılım fonksiyonu μ değeri 0 ve σ^2 değeri 1 olan normal dağılım ailesinin bir elemanıdır.



Şekil 2.7. GNDF grafiğine bir örnek

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.1)$$

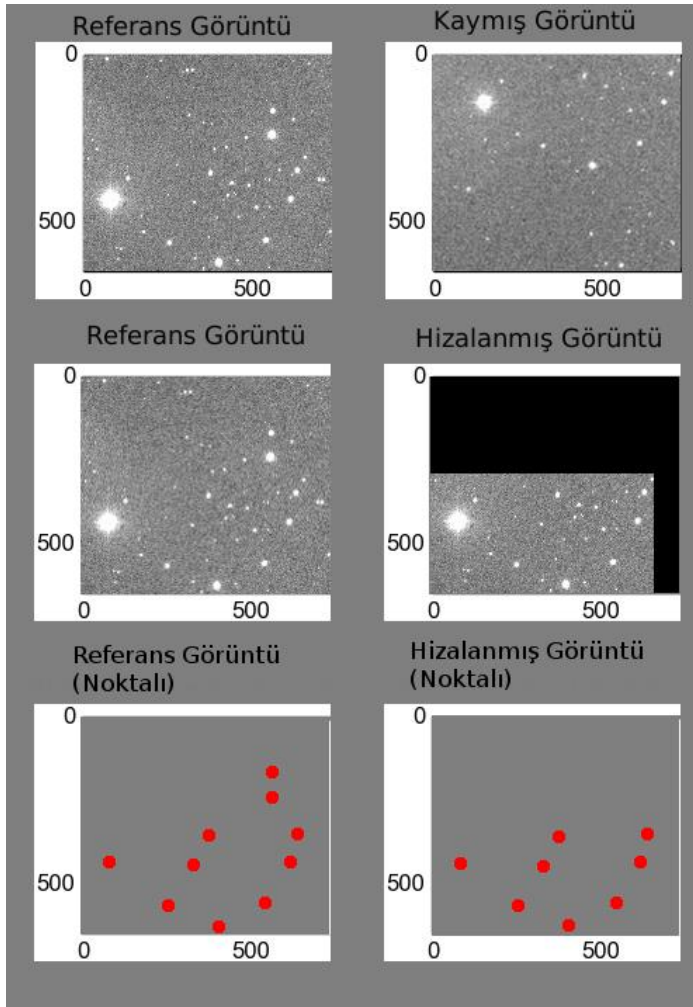
Denklem 2.1’de, μ aritmetik ortalama, σ ise standart sapma değerini gösterir (http://www.cihanatılmalı.com/mba/MBA_Konulari_Bolum_3_Istatistik.html). Bu formül, normal dağılım fonksiyonunun olasılık yoğunluk fonksiyonudur. GNDF’nun dağılım aralığı $(-\infty, +\infty)$ aralığındadır. Bu olasılık yoğunluk fonksiyonunun şeklini belirleyen en önemli değişken σ değeridir. Bu değer gauss eğrisinin varyans (değişme) değeri olarak kabul edilir. Ayrıca formülde bulunan μ değeri, ortalama değerdir. Gauss eğrisinin tepe noktasının hangi noktada değerinin maksimum düzeyde olduğunu belirler ve eğri bu değer etrafında simetrik. Bu denklemin geometrik görüntüsü Şekil 3.1.1 de görüldüğü gibidir. Örneğin, 3.6 noktası tepe noktasıdır, bu durumda $\mu=3.6$ olarak alınabilir. Dağılım fonksiyonlarında en belirgin özellik, fonksiyon ile çizilen eğrinin altındaki alan 1 değerinde olmalıdır.

Normal dağılımda μ ve σ değerinin her farklı değeri için farklı bir dağılım elde edilir. σ ‘nın farklı değerleri dağılımın eğrisinin şeklini belirleyecektir. Bu ilişkiyi bağıntılarda ifade edersek sistemin modeli yapılmış olur. Bu model ile sistemin sınır değerlerine göre oluşturulan bağıntılar ile yeni ölçümler yapmadan da bir bağıntının doğruluk derecesine göre diğer değişken hesaplanabilir. Hesaplamalarda değişkenlerden birisi bağımsız diğeri bağımlı değişken olarak tanımlanır. Bazen bağımlı değişkene tahmin edilen diğerine ise tahmin ettirici adı verilir. Bu bağıntılar eğriyi temsil eden fonksiyonlardır.

2.5. Görüntülerin hizalanması

Görüntülerin hizalanması işlemini **sources** adı verilen Python modülü içerisinde bulunan **align** isimli fonksiyon tarafından yapılmaktadır. Bu fonksiyon, **alipy** modülü içerisindeki **affineremap** isimli fonksiyonu çağırır ve kullanıcının belirleyebileceği bir görüntüyü referans olarak seçer, referans görüntü ile diğer görüntüler arasındaki AFİN Dönüşümü matrislerini hesaplar. Sonrasında bu matrisler kullanılarak tüm görüntüler hizalanacaktır. Bu hesaplama işlemi 3.2.1. bölümde detaylı olarak anlatıldı.

Hizalama işlemleri için, elde edilen referans katalog dosyasındaki görüntü ile diğer görüntülerin katalog dosyasındaki görüntüleri eşleştirmek için Lang ve arkadaşlarının 2010 yılında yapmış olduğu çalışma baz alınarak scipy afin dönüşümü fonksiyonu kullanılmıştır. Bu çalışmada görüntülere uygulanan afin dönüşümü Şekil 3.2.1’de aşamalar halinde gösterilmiştir.

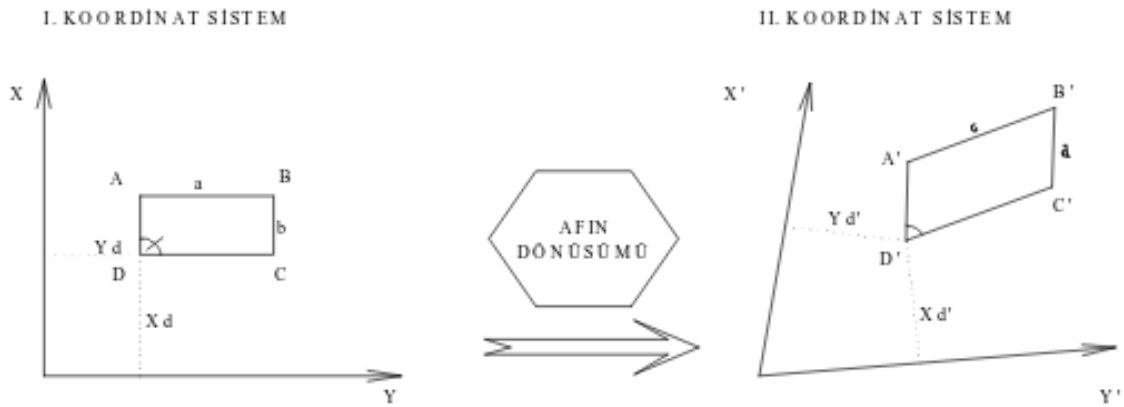


Şekil 2.8. Örnek görüntünün referans görüntü kullanılması ile hizalama (align) işleminin yapılma aşamaları

2.5.1 AFİN dönüşümü

Afin dönüşümü fotogrametri işlemlerinde tercih edilen bir yöntemdir. Afin dönüşümü üç boyutlu modellemelerin temel taşı denebilir ve bu alanda kullanılan dönüşümlerin en yaygın sınıfıdır. Aslında afin dönüşümler, uzaklıkların ve eşdoğrusallığın korunmuş olduğu dönüşümlerdir. Örneğin, belirli bir çizgideki orta nokta çizginin orta noktası olarak kalır ve düz bir çizgide bulunan tüm noktalar, bir afin dönüşümü uygulandıktan sonra aynı kalacaktır (Anonim 2016). Açılar dönüşümden sonra değişir ve bu değişim açı kollarının doğrultusuna bağlıdır. Herhangi bir doğru veya paralel formu bozulmadan aynı kalmaya devam edecektir. Örneğin, bir kare afin dönüşümü sonucu paralel kenara dönüşmektedir (Şekil 2.9). Afin dönüşümü, dik ve eğik koordinatlar arasındaki bir dönüşüm bağıntısıdır. Afin dönüşümünde eksenler yönünde farklı ölçek ve dönüklük parametreleri vardır. Bu yüzden açı koruma özelliğine sahip değildir. Görüntüyü sayısallaştırırken eksen deformasyonları (çevresel, aletsel etkiler) oluştuğu için afin dönüşümü uygun bir çözüm sağlamaktadır.

Afin dönüşümünde koordinat eksenleri yönündeki birimler ayrı olmaz. Uzunluklar ve açılar dönüşüm yönüne bağlı olarak değişecektir. Ancak açıların değişimi açı kollarının doğrultusuna bağlıdır. Alanlar dönüşümden sonra az miktarda etkilenir. Bu miktar dönüşüm matrisinin determinantına eşit olacaktır.



Şekil 2.9. Afin dönüşümünde eksenler yönünde kayma ve dönme örneği (Ünsal,2009)

Bu dönüşümde eksenler yönünde iki ölçek faktörü, ikişer öteleme ve dönme olmak üzere sonuçta altı parametre etkilidir. Bu parametrelerin çözümü için x-y ve u-v koordinat sistemlerinde bilinen üç noktaya ihtiyaç vardır. Ortak nokta sayısının üç ve daha fazla olması durumunda altı parametre en küçük kareler yöntemine göre hesaplanır.

Afin dönüşümünde iki koordinat sistemi arasındaki ilişki ise;

$$x = ax + by + c \quad (2.2)$$

$$y = dx + ey + f \quad (2.3)$$

denklemleriyle tanımlanır. Açılar ise dönüşümden sonra değişime uğrar. Çözüm oluşturulması için altı katsayı belirlenmelidir. Bu katsayılar 2.2 ve 2.3 denklemlerinde geçen a, b, c, d, e, f katsayılarıdır.

Afin dönüşümler tipik olarak bir dönüşüm matrisi M ve onun tersi M^{-1} yoluyla uygulanır. Örneğin, üç boyutlu uzayda bir noktaya bir afin dönüşümü uygulamak için tanımlanan P noktasını, Q noktasına dönüştürmek için aşağıdaki denklemi alırız (Anonim 2016).

$$Q = MP \quad (2.4)$$

Bu denklem sistemi genişletilmiş formda şu şekilde ifade edilir;

$$\begin{aligned} x_1 &= a_0 + a_1 \cdot u_1 + a_2 \cdot v_1 \\ y_1 &= a_3 + a_4 \cdot u_1 + a_5 \cdot v_1 \\ &\dots \\ x_n &= a_0 + a_1 \cdot u_n + a_2 \cdot v_n \\ y_n &= a_3 + a_4 \cdot u_n + a_5 \cdot v_n \end{aligned} \quad (2.5)$$

şeklinde oluşur.

Matris şeklinde gösterimi;

$$\begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix} \begin{bmatrix} 1 & u_1 & v_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & u_1 & v_1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & u_n & v_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & u_n & v_n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (2.6)$$

Yani, Denklem 2.4 lineer denklem sistemi şeklinde çözülebilir. Bu denklem sisteminin çözümü olması için $N=3$ olmalıdır. $N > 3$ ise sistem dengelemeli çözülebilir (N = bilinen değer sayısı). Çözüm ise;

$$L + v = A \cdot x \quad (2.7)$$

sistemi ile yapılır. Bu sistemi en küçük kareler yöntemi ile çözebiliriz. Bunun için;

$$v = A \cdot x - L \quad (2.8)$$

ve,

$$x = (A^T \cdot A)^{-1} \cdot A^T \cdot L \quad (2.9)$$

dengelemeli çözüm bağıntısıyla çözülür.

Dengelemeli çözüm, denklem sayısının bilinmeyen sayısından çok olduğu durumdur. En küçük kareler yöntemine göre dengelemeli çözüm için x ve y eşitliklerinden yararlanılarak;

$$F(u, v, x, y) = a_0 + a_1(u + v_u) + a_2(v + V_v) - (x - V_x) = 0 \quad (2.10)$$

$$G(u, v, x, y) = a_3 + a_4(u + v_u) + a_5(v + V_v) - (y - V_y) = 0 \quad (2.11)$$

(x,y) uzayından (u,v) uzayına dönüşüm gerçekleşir. Bu şekilde görüntülere uygulandığında hizalama işlemi için kullanılabilir.

2.5.2. En küçük kareler yöntemi

Günlük hayatta çeşitli alanlarda herhangi bir uygulama ile toplanan veriler tablo halinde incelenebilir ve bu tabloya uygun bir fonksiyon bulunmaya çalışılır. Bir veri tablosuna en iyi uyan fonksiyonu bulmaya regresyon analizi denir. Analizde değişkenler arasındaki ilişkiyi veren fonksiyon;

$$Y = f(X) \quad (2.12)$$

ile gösterilir.

Burada Y değişkeni bağımlı, açıklanan veya etkilenen değişken olarak adlandırılır. X değişkeni ise buna paralel olarak bağımsız, açıklayıcı veya etkileyici değişkendir. Bu iki değişkenin arasındaki ilişki doğrusal ise;

$$Y_i = \beta_0 + \beta_1 X_i \quad (2.13)$$

olarak yazılabilir.

Eğer değişkenler arasında doğrusal olmayan bir ilişki varsa;

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 \quad (2.14)$$

şeklinde ifade edilebilir.

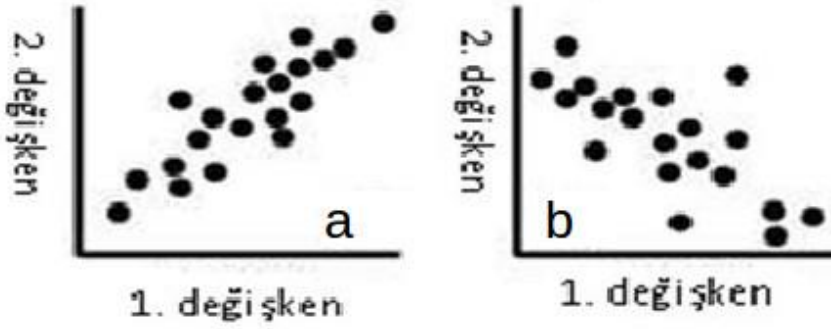
Burada amaç β_0 ve β_1 parametrelerini tahmin etmektir. β_0 ve β_1 bilinmeyen regresyon katsayılarıdır.

β_0 : doğrunun y eksenini kestiği nokta

β_1 : doğrunun eğimi

İki değişkenli ve doğrusal regresyon denklemi yani basit doğrusal regresyon denkleminde en küçük kareler yöntemi denir. Regresyon analizinde çok tercih edilen yöntemdir. En küçük kareler yönteminde verinin eğilimine bağlı olarak Şekil 2.10'daki gibi pozitif veya negatif yönde dağılım olabilir.

Regresyon modelinin belirlenmesinde serpilme diyagramından yararlanılabilir. Y ve X değişkenine ilişkin elde edilen verilerden hareketle çizilen serpilme diyagramı aşağıdaki gibi olduğunda regresyon denkleminin doğrusal, yani 2.13 denkleminin formunda olması beklenir.



Şekil 2.10.a) Pozitif doğrusal kuvvetli ilişkili b) Negatif doğrusal kuvvetli ilişkili

3. MATERYAL VE METOT

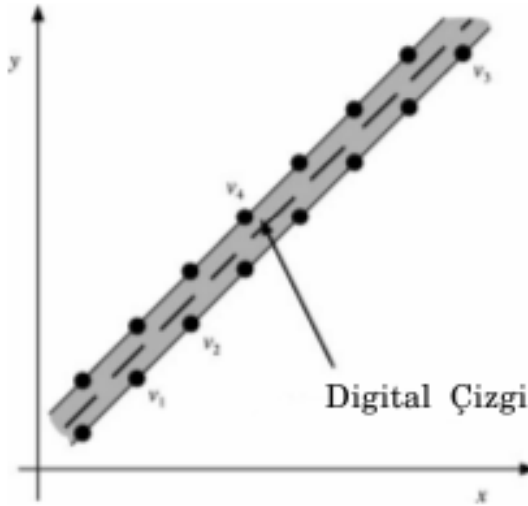
Keşfedilen asteroid sayısı son yıllarda artmaya devam etmektedir. YYA keşifleri dünyamız açısından fazlasıyla önem arz etmektedir. Asteroidlerin çeşitli tespit yöntemlerinden Giriş bölümünde bahsedildi. Bu çalışmada ise herhangi bir teleskop ile alınan görüntülerle çalışabilen açık kaynak kodlu bir program tasarlandı (Atay vd., 2016). Bu programın amacı, 20 kadir parlaklığa kadar ve çeşitli poz sürelerinde alınmış teleskop görüntülerinden otomatik olarak hareketli cisim tespit edebilmektir. Hareketli cisimlerden özellikle asteroidler için tasarlanan A-Track programında kullanmak için klasik Doğru Tanıma Yönteminden (DTY) geliştirilen Çoklu Görüntüden Doğru Tanıma Algoritması (Multiple Image Line Detection, MILD) kullanıldı. MILD algoritması, daha önce astronomi alanında kullanılmamış olan klasik DTY temel alınarak teleskop görüntülerine uygulanan bir yöntemdir.

3.1. Klasik Doğru Tanıma Yöntemi

Birçok alanda kullanılan doğru tanıma algoritmasında, öncelikle iki nokta ele alınır, daha sonra üçüncü bir nokta alınıp bu iki nokta ile doğrudaş olup olmadığı kontrol edilir. Eğer alınan üç noktanın doğrudaş olduğu doğrulanırsa bu karşılaştırma ana görüntüdeki diğer noktalar için de tekrarlanır.

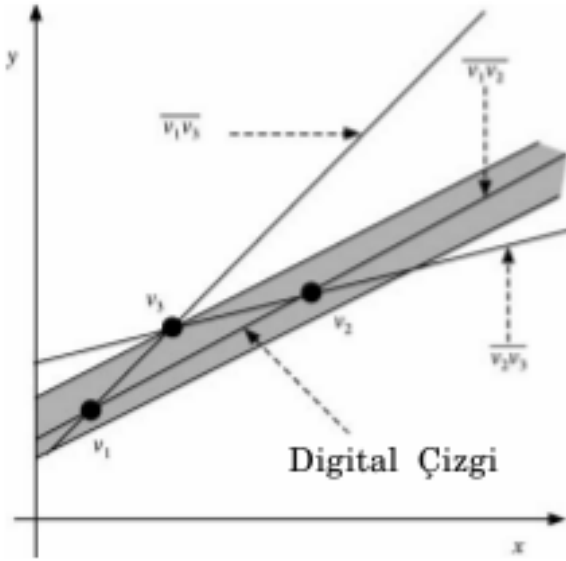
Doğru tespit etme yöntemi temelde tek bir görüntü üzerindeki noktaları tespit etmek için kullanılır. Doğru oluşturabilecek tüm noktalar değerlendirilmektedir.

Herhangi üç noktanın doğrudaş olup olmaması için 3 şart vardır:



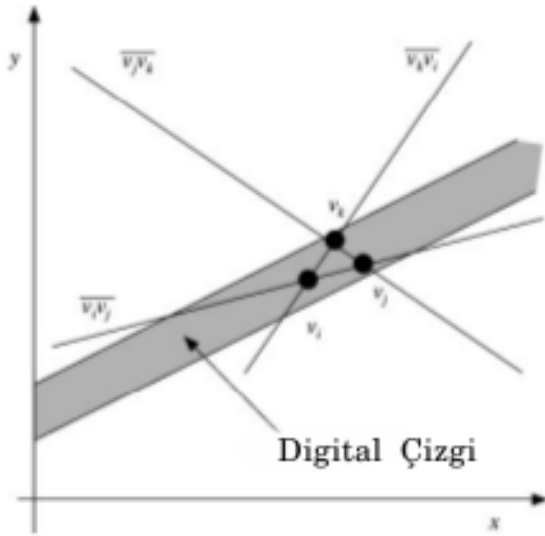
Şekil 3.1. DTY'nin doğrudaşlık için birinci şartı

1. Taban (bir diğer noktanın en uzak hangi mesafede olacağını belirlemek için) (Şekil 3.1)



Şekil 3.2. DTY'nin doğrudaşlık için ikinci şartı

2. Yükseklik (belirlenen bir sınır değerinden daha düşük olmalı) (Şekil 3.2)



Şekil 3.3. DTY'nin doğrudaşlık için üçüncü şartı

3. Alan (belirlenen alan değerinden daha büyük olmalı) (Şekil 3.3)

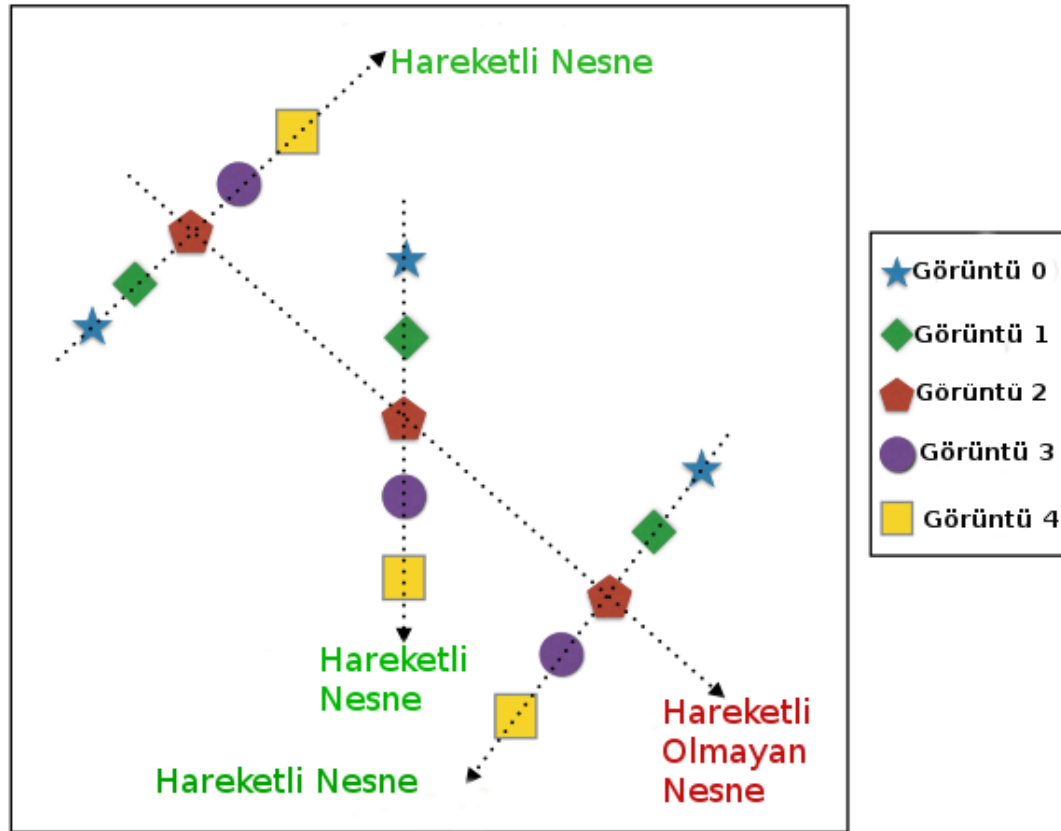
Bu şartları sağlayan herhangi üç noktanın doğrudaş olduğu kabul edilir.

Burada amaç, geliştirilen koşullara göre, tespit edilen cisimlerden aynı cisme ait noktalardan bir doğruyu tespit etmektir. Ancak klasik DTY hareket halindeki cisimlere ait noktaların tek bir cisimden gelip gelmediğini kontrol eden bir yöntem değildir ve hareketli cisim tespitinde kullanılmamıştır. Bu yüzden astronomik amaçlı alınmış görüntülerde kullanılabilmesi için DTY'nin geliştirilmesi gerekmektedir.

3.2. Çoklu Görüntüden Doğru Tanıma Algoritması (Multiple Image Line Detection (MILD))

Geliştirilen Multiple Image Line Detection (Çoklu Görüntülerden Doğru Tespiti) (MILD) yönteminde Randomized Line Detection (RLD)(Chen ve Chung, 2001) yönteminden esinlenerek, ardışık alınmış teleskop görüntülerine uygulayabilmek için doğrudanlık şartları tekrar belirlendi. MILD yöntemi hizalanmış ve kataloglanmış olan teleskop görüntülerine ihtiyaç duyar.

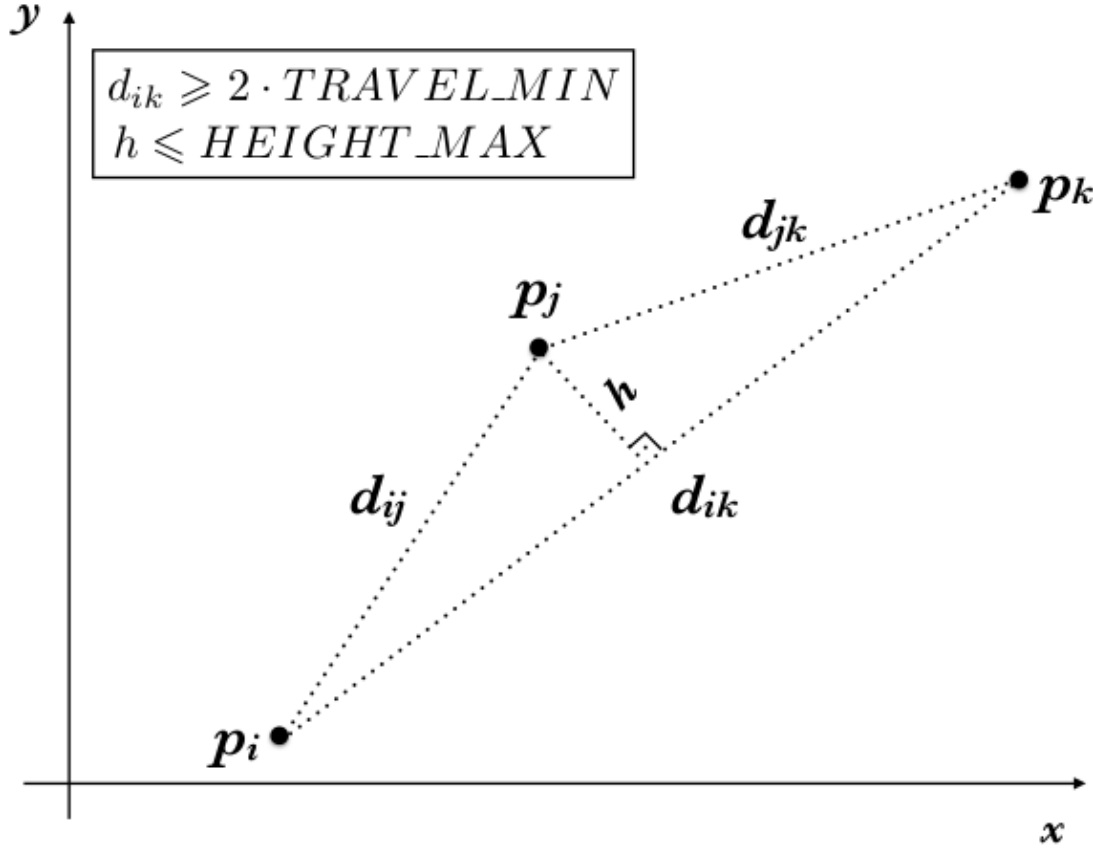
MILD yönteminin ilk önemli özelliği doğrudanlık kontrolüne tabi tutulan noktaların farklı görüntülere ait olması şartının uygulanmasıdır. Daha sonra en az üç görüntüden birer aday nokta seçilmesi ve bu üç noktanın aynı doğruyu oluşturup oluşturmadığının belirlenmesidir. İkinci önemli özelliği ise, hareketli cismin belirlenmesinde aday cismin hızını tahmin ederek kullanılmasıdır.



Şekil 3.4. MILD'in doğrudanlık için birinci şartı

MILD yönteminde Şekil 3.4 'de görüldüğü gibi, her şekil farklı bir görüntüyü temsil etmekte olup toplamda örnek 5 görüntü (3 ile 15 görüntü en ideal aralıktır) vardır. Bu 5 görüntü üst üste bindirildiğinde oluşan görüntüye bakıldığında, MILD yönteminin Şekil 3.4'te oluşan görüntüde 3 hareketli cisim tespit ettiği görülür. Burada 3 noktanın oluşturduğu doğrunun üzerindeki noktaların her biri farklı bir görüntüden

gelmektedir. Ancak MILD 1 tane hareketli olmayan nesneye ait doğruyu rededecektir. Doğrunun üzerindeki noktalar aynı görüntüden gelen ve rastlantısal olarak doğru oluşturduğundan hareketli nesne olarak kabul edilmeyecektir. Tesadüf eseri aynı görüntüye ait noktalar bir doğru oluşturabilir. Ancak bu noktaların oluşturduğu doğru hareketli bir cisme ait olmayacaktır. MILD’da amaç aynı doğru üzerinde farklı görüntülerden gelen noktaları tespit etmektir.



Şekil 3.5. MILD’ın doğrudaşlık için ikinci şartı

Şekil 3.5.’de gösterilen p_i , p_j , p_k gibi aday noktalar üçlü kombinasyonlar halinde gruplanır ve teker teker her üçlüye bakılır. Her üçlünün oluşturduğu üçgenin en uzun kenarının belirli bir sınır değerden büyük olması beklenir ve bu değer $TRAVEL_MIN$ parametresi ile gösterilmiştir. Bu değer d_{ik} kenarını temsil eder.

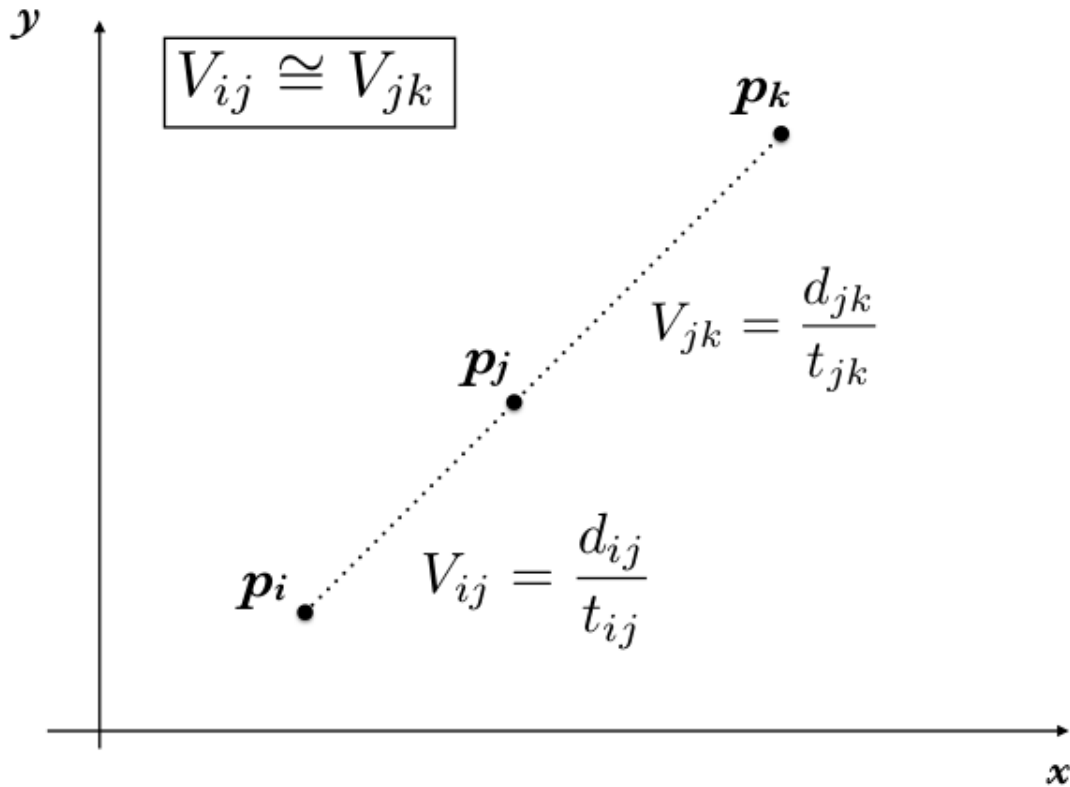
Daha sonra $TRAVEL_MIN$ değerini sağlayan üçlünün oluşturduğu üçgenin en uzun kenarına inen yüksekliğinin belirli bir sınır değerden küçük olması beklenir. Kod içerisinde bu sınır değeri $HEIGHT_MAX$ parametresi ile gösterilmiştir.

En uzun kenar şartının belirlendiği $TRAVEL_MIN$ parametresinin hesaplanması için görüntünün master dosyasındaki her üçlü nokta kombinasyonunun konum bilgisi kullanılır.

İkinci olarak, bu üçgenin, en uzun kenarına inen yüksekliğinin sıfıra yakın olup olmadığı kontrol edilir. i, j, k noktalarından her biri için o noktaya karşılık gelen $d_{k \rightarrow ij}$ yüksekliği şu şekilde bulunur:

$$d_{k \rightarrow ij} = \frac{((x_j - x_i)y_k + (y_i - y_j)x_k + x_i y_j - x_j y_i)}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \quad (3.1)$$

Bu şekilde, çok sayıda teleskop görüntüsüne ihtiyaç duymadan, herhangi bir teleskop sistemine bağımlı olmadan, görsel doğrulamaya gerek bırakmadan ve şimdiye kadar bu alanda kullanılmamış bir yöntemle güneş sistemindeki hareketli cisimleri otomatik olarak tespit edecek bir program tasarlandı.



Şekil 3.6. MILD'ın doğrudanlık için üçüncü şartı

MILD ile klasik DTY'deki alan sınırlama şartına artık ihtiyaç duyulmaz. Bunun yerine, aynı hareketli nesneye ait olduğu düşünülen i, j, k noktalarının gözlem zamanları kontrol edilerek, bu hareketli cismin sabit hızla hareket edip etmediğine bakılır. Asteroidlerin kısa mesafeler arasında, bir doğru üzerinde sabit hızla hareket ettikleri varsayılır. Bu varsayıma göre, Şekil 3.6'daki gibi p_i ve p_j noktaları arasındaki hızın p_j ve p_k noktaları arasındaki hıza eşit olması gerekir. Dolayısıyla, i ve j noktaları arasındaki mesafe d_{ij} , geçen zaman da Δt_{ij} ile gösterilirse;

$$V_{asteroid} \simeq \frac{d_{12}}{\Delta t_{12}} \simeq \frac{d_{23}}{\Delta t_{23}} \quad (3.2)$$

eşitliğinin sağlanması beklenir.

Farklı görüntülerden gelen ve bu kriterleri sağlayan noktaların aynı asteroide ait olması beklenir. Bu şekilde, eldeki görüntülerdeki bütün üçlü kombinasyonlar tek tek incelenerek hangi kaynak noktaların asteroid olduğu tespit edilir, bu noktalara ait bilgiler (görüntü numarası, koordinatlar, görseller, vb.) bir rapor haline getirilir. Bu raporlama işlemi sonucunda sonuçlar A-Track yazılımının çalıştırıldığı dizinde oluşturulacak olan /A-Track klasörüne .txt formatında kaydedilir.

3.2.1. A-Track programı

MILD algoritmasını ardışık teleskop görüntülerine uygulayabilmek için A-Track adında bir program yazıldı. Bu program ile dışarıdan müdahaleye çok gerek duymadan gerekli konfigürasyon parametreleri girilerek istenilen sonuçların alınması amaçlandı.

A-Track programı ile nesne tespiti iki aşamadan oluşuyor. İlk aşama işlenmek üzere verilen görüntüdeki sabit cisimlerin tespiti ve sonraki aşama hareketli cisimlerin tespit edilmesidir. Bu çalışmada geliştirilen algoritma dört aşamadan oluşur:

1. Görüntülerin hizalanması.
2. Görüntülerdeki ışık kaynaklarının tespiti.
3. Işık kaynakları arasından asteroid adaylarının belirlenmesi (MILD).
4. Asteroid adayları arasından gerçekten asteroid olan cisimlerin belirlenmesi ve raporlanması (A-Track).

1 ve 2. aşamalarda görüntüler hizalanacak ve mevcut ışık kaynakları tespit edilecektir. Hizalama için alipy³ modülünden, kaynak tespiti için de Sextractor^{4,5} yazılımından yararlanılacaktır. 3. aşamada, bulunan kaynaklar yani aday noktalar uygun parametreler kullanılarak filtrelenir. Bu parametreler görüntülerin header bilgilerinden okunur ve bölüm 3.2’de bahsedilen aşama tamamlanır.

İlk aşamada hareketli olmayan cisimlerin elenmesi işi, A-Track programının **asteroids** adı verilen Python modülü içerisinde bulunan **detect_candidates** isimli fonksiyon tarafından yapılması sağlanır. Bu aşamada kullanılan parametreler konfigürasyon dosyasının [asteroids] kısmındadır. Hareketli cisimleri tespit etmek için kodun bu kısmında yüksek işlemci gücü gerektiğinden Python3 programlama diline özgü paralel hesaplama araçları (*Multiprocessing* modülü) kullanılmıştır.

Bu aşamadan sonra, her bir görüntüde geriye kalan cisimler aday cisimler olarak etiketlenip görüntü adı ve **.cnd** uzantısı ile atrack/ dizinine bir katalog dosyası olarak kaydedilir. Bu dosyalarda, görüntülerde belirlenen her bir aday cismin dahili işaret kodu (FLAGS), x ve y koordinatları (X_IMAGE, Y_IMAGE), akısı (FLUX_AUTO) ve yerel arkaplan değeri (BACKGROUND) bulunmaktadır. Aday cisimler arasında, gerçekten hareketli cisimlerin yanı sıra kozmik ışımlar ve kötü atmosferik koşullar sebebi ile (merkez noktalarındaki kaymalardan dolayı) hareketli gibi görünen hareketsiz cisimler (yıldızlar) de olabilir.

Aday cisimlerin belirlenmesi altı koşulun kontrol edilmesi ile sağlanır. İlk olarak, cismin dahili işaret kodları kontrol edilir. Bu kodlar Bölüm 2.3.1’de detaylı olarak anlatıldı. Bir cismin aday cisim olabilmesi için dahili işaret kodunun 16’dan

küçük olması şartı aranır. Aday cismin dahili işaret kodu yani FLAG değeri yükseldikçe görüntüde cismin ayırt edilmesi zorlaşacaktır.

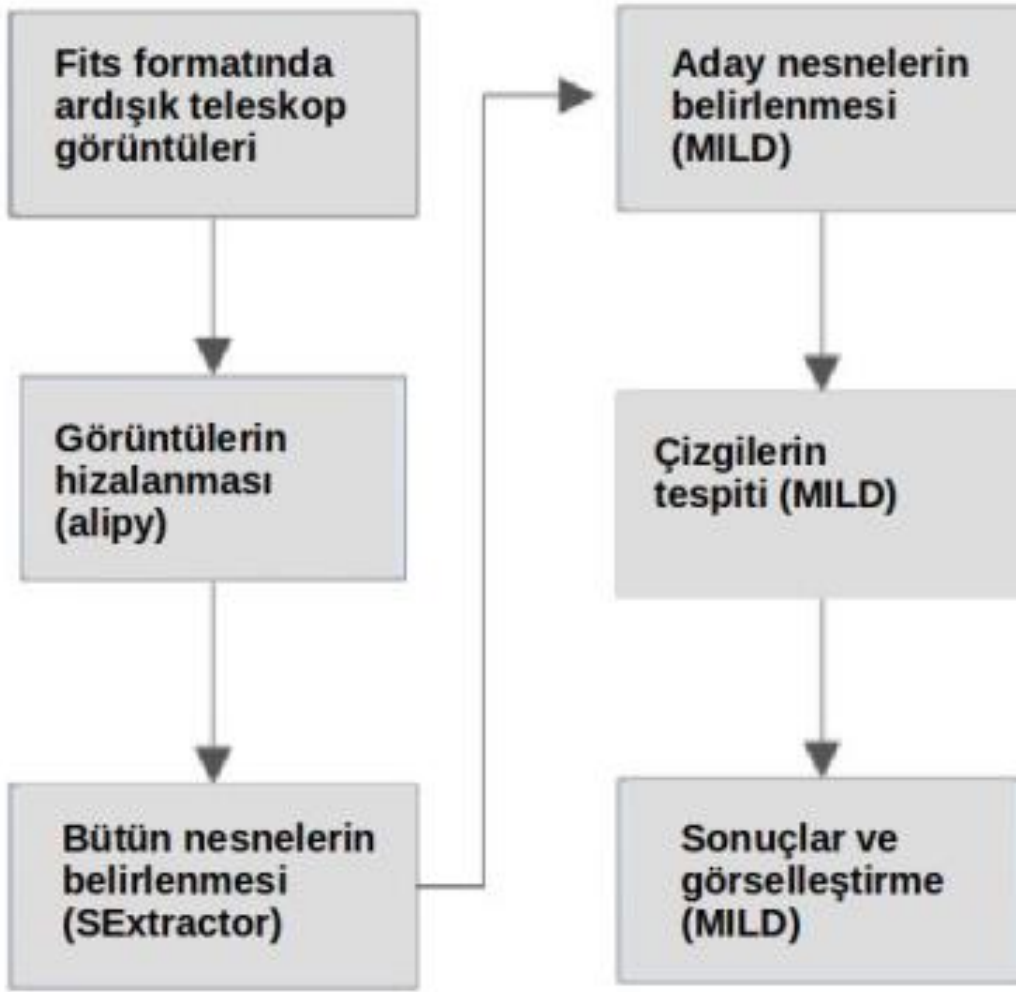
İkinci olarak, cismin yarı-doruk genişliği (FWHM) kontrol edilir. Bu değerin, konfigürasyon dosyasındaki FWHM_MIN değeri ile bütün görüntülerdeki ortalama yarı-doruk genişliğinin ampirik olarak belirlenen iki buçuk katı arasında olması şartı aranır.

Üçüncü olarak, cismin akı (FLUX) değerinin, yerel arkaplan şiddeti ile konfigürasyon dosyasındaki FLUX_MAX değeri arasında olması şartı aranır.

Dördüncü olarak, cismin sinyal-gürültü oranının (SNR) konfigürasyon dosyasındaki SNR değerinden büyük olması şartı aranır.

Beşinci olarak, cismin uzama (ELONGATION) değerinin konfigürasyon dosyasındaki ELONGATION_MAX değerinden küçük olması şartı aranır.

Son olarak, cismin x ve y koordinatlarının, konfigürasyon dosyasının asteroids kısmında bulunan TRAVEL_MIN isimli tolerans değeri göz önünde bulundurularak, farklı görüntülerde değişip değişmediği kontrol edilir. Bunun için, önce, ışık kaynaklarının belirlenmesi aşamasında kaydedilmiş olan .pysexcat uzantılı katalog dosyalarının birleşiminden **master.pysexcat** isimli genel bir katalog dosyası oluşturulup atrack/ dizini içine kaydedilir. Sonra, cisim ile bu katalog dosyasında yer alan diğer bütün cisimler arasındaki uzaklıklar hesaplanır. Hesaplanan uzaklıkların TRAVEL_MIN tolerans değerinden küçük olduğu durumlar sayılır. Bu sayı 1 ise, yani cismin koordinatları her görüntüde değişiyorsa, cisim, aday cisim olarak etiketlenir.



Şekil 3.7. A-Track programının çalışma şeması

Şekil 3.7’de gösterildiği gibi A-Track programı beş dosyadan oluşur. Ancak 2.modül olan *sources.py* hem cisimlerin hizalanması hem de cisimlerin tespitinde kullanılır;

1. *atrack.py*, komut-satırından çalıştırılan ana program dosyasıdır.
2. *sources.py*, görüntülerin hizalanması ve cisimlerin tespiti için kullanılan modüldür. Bu modülün kodlanmasında, teleskop görüntülerinin hizalanması için geliştirilmiş olan *alipy* ve teleskop görüntülerindeki cisimlerin tespiti için geliştirilmiş olan *SExtractor* paketlerinden yararlanılmıştır. Bu paketlerin her ikisi de açık-kaynak kodludur.
3. *asteroids.py*, MILD algoritmasının bulunduğu, aday cisimlerin ve sonrasında da hareketli cisimlere ait doğruların tespit edildiği modüldür. Bu modüldeki aşamalardan biri olan hareketli cisimlerin tespit edilmesi aşaması, yüksek işlem gücü gerektirdiğinden, daha önce bahsedilen uygun paralel programlama modülleri kodlanmıştır.

4. *visuals.py*, görselleştirme işlemlerini gerçekleştiren modüldür. Programın görsel çıktıları bu modül tarafından oluşturulur.
5. *atrack.config*, program parametrelerinin saklandığı konfigürasyon dosyasıdır. A-Track tarafından kullanılan bütün parametreler ve bu parametrelerin açıklamaları bu dosyada bulunur. Kullanılan ön tanımlı değerlerin bir kısmı FITS görüntülerinin alındığı teleskop/CCD sistemine bağlıdır. Diğerleri ampirik değerlerdir.

MILD yöntemi kullanılarak tasarlanan bu yazılımı diğerlerinden ayıran iki önemli fark vardır. Birincisi, diğer programlar gibi ücretli ve kapalı kaynak kodlu olmaması ikincisi ise A-Track programını diğerlerinden ayıran en önemli özellik olan hızı ve fazla sayıda asteroid tespit edebilmesidir. Hızının sebebi A-Track yazılımının Python paralel programlama kütüphaneleri kullanılarak bilgisayarın yüksek CPU gücünden faydalanılmasıdır.

4.BULGULAR VE TARTIŞMA

4.1. Kullanılan Görüntülerin Özellikleri

Tezin bu aşamasında TUG TÜBİTAK-14BT100-648 kodlu ve TÜBİTAK - 14BT100-327 proje kapsamında T100 teleskobu ile alınmış asteroid gözlem verileri kullanılmıştır. Kullanılan veriler arasında temiz yani görüntülerde gözlem sırasında büyük ve parlak yıldız olmayan, görüntü kalitesinin iyi olduğu ve zaman kaymasının az olduğu, header bilgileri tam girilmiş veriler seçilmiştir. Kullanılan bu verilerin farklı zaman ve mevsimlerde alınmış olmasına önem verilmiştir. Bir amaç da programı farklı mevsim şartlarında alınmış görüntülerle test etmektir. Aynı zamanda, atrack.config dosyasındaki ampirik değerlere de bu verilerden elde edilen sonuçlar analiz edilerek ulaşılmıştır.

Ek olarak, astreroid gözlemlerinde gözlemlenecek cismin parlaklığına ve hızına dikkat edilmelidir. Gözlem yapılacak aletin özellikleri gözlenecek cisim için uygun olmalıdır. Yani, gözlenecek cismin parlaklık derecesinin teleskobun ölçüm sınırına uygun olması ve gözlem sırasında vermiş olduğumuz poz süresinin cismin hızına göre ayarlanmış olmasıdır. Aksi takdirde görüntülerde uzamalar meydana gelir. Bu uzamalar da ışık kaynaklarının merkez koordinatlarının elde edilmesinde hatalara yol açar.

Bu tez çalışmasında iki farklı proje dahilinde 2012 ve 2014 senelerinde zaman, mevsim, hız, parlaklık bakımlarından çok çeşitliliğe sahip verilerle çalışıldı. Komut satırından çalıştırılan bu programda kullanılan veri setleri 2500 metre yükseklikteki TÜBİTAK Ulusal Gözlemevi'nde (TUG) bulunan 100 cm çaplı T100 teleskobuna bağlı Spectral Instruments (SI) 1100 4096 x 4037 px, CCD'sinden elde edilmiştir. Yazılımın performansını denetleyebilmek adına veri setlerinin SNR değerlerinin düşük, orta ve yüksek olmasına ve çok kaynak bulunan gözlem verilerin seçilmesine özen gösterilmiştir. Seçilen görüntülerde header bilgilerinde, algoritmanın çalışmasında önemli olan, gözlem zamanı (DATE-OBS), alınan görüntünün poz süresi (EXPTIME), binning miktarı (XBIN, YBIN), boyutu (NAXIS1, NAXIS2) değerlerine sahip olması gerekmektedir.

TÜBİTAK -14BT100-648 kodlu proje ile gözlem yapılan cisimler 100- 200 EXPTIME süresine sahiptir. Bu verilerde gözlenmesi hedeflenmiş cisimler genellikle hızlı hareket eden cisimlerdir.

TÜBİTAK -14BT100-327 kodlu proje ile gözlem yapılan cisimler 200-300 EXPTIME süresine sahiptir. Bu verilerde ise gözlenmesi hedeflenmiş cisimler genellikle yavaş hareket eden cisimlerdir.

Her iki projede de Tübitak Ulusal Gözlemevi'ndeki (TUG) T100 gözlem sonuçlarıdır. T100 teleskobunun CCD özellikleri bölüm 2.1.2' de anlatılmıştır.

4.2. Benzer Programlar İle Karşılaştırılması

Karşılaştırma sonuçlarında, temelde aynı amaca yönelik farklı algoritmalarla yazılmış olan üç program Astrometrica, CoLiTeC(LookSky) ve A-Track programları karşılaştırıldı. Astrometrica ve CoLiTeC programları arayüz ile çalışmakta ancak A-Track komut satırından çalışmaktadır. Çalışma sürelerini karşılaştırabilmek için bütün bu testlerde aynı bilgisayar kullanılmıştır.

Çizelge 4.1.'de görüldüğü gibi, Astrometrica programı için tüm bu işlemler yaklaşık 8-10 dakika mertebesinde sürmektedir. 6.sütun görüntülerde olması gereken asteroid sayısından ne kadarının tespit edilebildiği, 7.sütun ise yanlış tespit edilen cisim sayısını göstermektedir. CoLiTeC programı için ise tüm bu işlemler 1-3 saat mertebesinde sürmektedir ve tablonun 8. sütunu görüntülerde olması gereken asteroid sayısından ne kadarının tespit edilebildiği, 9.sütun ise yanlış tespit edilen cisim sayısını göstermektedir. Görüntülerde olması beklenen asteroid sayısı AsPlot web uygulaması ile ilgili görüntünün koordinat değerleri ve gözlem günü, saati girilerek elde edilen sonuçlara göre belirlendi. Tabloda açıkça A-Track uygulamasının daha çok sayıda asteroid tespit ettiği görülmektedir. Ek olarak diğer iki programa göre çok daha hızlı çalıştığı görülmektedir.

Çizelge 4.1. Tablonun süre sütununda 1.veri A-Track, 2. Veri Astrometrica, 3.veri Colitec çalışma sürelerini göstermektedir

VERİ İSİMLERİ	RA DEC	GÖZLEM ZAMANLARI	A-TRACK DOĞRU TESPİT	A-TRACK YANLIŞ TESPİT	ASTROMETRİCA DOĞRU TESPİT	ASTROMETRİCA YANLIŞ TESPİT	ColiTeC DOĞRU TESPİT	ColiTeC YANLIŞ TESPİT
11922	04 00 59.4 22 50 45	Tarih : 2013-12-06 Zaman : 01:34:06.16	3	1	4	1	0	0
2000 EV136	09 12 19.3 20 14 05	Tarih : 2014-02-06 Zaman : 01:13:51.47	4	1	3	1	0	0
7283	12 26 33.1 02 37 37	Tarih : 2015-04-13 Zaman : 20:27:17.44	4	0	2	0	0	0
79213	21 40 09.4 18 06 29	Tarih : 2015-08-05 Zaman : 22:27:11.58	2	0	0	0	0	0
7719	06 32 23.9 20 26 46	Tarih : 2014-11-03 Zaman : 21:02:39.12	1	4	0	0	0	1
Arago	05 56 15.2 42 31 25	Tarih : 2015-03-08 Zaman : 17:56:34.71	2	0	1	1	0	0
30192	17 18 26.5 13 48 50	Tarih : 2015-08-18 Zaman : 21:07:15	5	7	0	0	2	> 15
33915	20 39 08.1 08 18 37	Tarih : 2014-07-27 Zaman : 19:35:46.29	0	0	0	0	5	Çok sayıda
89187	21 10 52.1 07 02 22	Tarih : 2014-07-27 Zaman : 19:11:40.82	2	3	1	2	3	9
2000 CA30	05 47 52.2 20 41 24	Tarih : 2014-02-05 Zaman : 18:31:33.85	7	0	0	0	0	2
43288	14 52 58.7 07 12 50	Tarih : 2014-05-11 Zaman : 20:54:30.61	13	0	2	0	3	5
TOPLAM SÜRE			~1 DAKİKA		~8 DAKİKA		~1 SAAT	

4.3. A-Track Performans Testi

Değişimleri kullanılan diğer parametrelere göre daha önemli sonuçlar veren dört parametre bulunmaktadır. Bunlar; HEIGHT_MAX, TRAVEL_MIN, SNR_MIN ve TOLERANCE'dir. Bu parametrelerden alt bölümlerde detaylı olarak bahsedildi. Çizelge 4.2., Çizelge 4.3., Çizelge 4.4., Çizelge 4.5.'de görülüyor ki ,bu dört parametrenin ön tanımlı değerleri sırasıyla; 0.15,0.5,10,1.0 'dır. Birçok veri setiyle yapılan testler sonucunda elde edilen bu ön tanımlı değerlerin alınmasının iki sebebi vardır. Bunlar;

- Özel bir cisim olmadığı sürece bulunmak istenen hareketli cisimlerin hızlarının belirli aralıklarda olması,
- Hareketli cisimlerin parlaklık değerlerinin kullanmış olduğumuz teleskoba bağlı olmasıdır.

Bu dört parametreye ait tablolardaki asteroid sayıları; AsPlot asteroid sayısı, AsPlot web uygulaması ile ilgili görüntünün koordinat değerleri ve gözlem günü, saati girilerek elde edilen sonuçlara göre belirlendi. A-Track doğru ve yanlış değerleri, görüntüde tespit edilen hareketli cisimlerden AsPlot asteroid sayısına göre doğru tespit edildiğinde A-Track doğru ve yanlış tespit varsa A-Track yanlış sütununa yazıldı. Belirsiz nesne olarak nitelendirilen cisimler ise gerçekten hareket edip etmediği kullanıcı tarafından görüntülerin hareketli gif formatında dosyalar incelenerek belirlenmesi gereken cisimlerdir.

A-Track programının performans testi için iki veri seti tercih edildi. Bu veri setlerinin isimleri ise ilk olarak 2000CA30, kalabalık bir asteroid bölgesinde, çok sayıda, hızlı hareket eden asteroidler bulunduğundan tercih edildi. İkinci veri seti: Astronomia, temiz bir alanda, az sayıda, yavaş hareket eden asteroidler bulunduğundan tercih edildi. Bu iki veri seti de aynı sayıda gözlem sayısı ve aynı parametre değerlerinde, aynı bilgisayar kullanarak test edildi. Veri setlerinde ön indirgeme yapılmadan işlemler yapıldı çünkü ön indirgeme yapıldığında da benzer sonuçlar alındı. Bu yüzden daha hızlı işlem yapılması için ön indirgeme yapılmayan ham görüntüler tercih edilmiştir (Atay vd., 2016).

4.3.1. HEIGHT_MAX parametresi

HEIGHT_MAX parametresinin değeri arttıkça gerçekleşmesi gereken iki kurala göre görüntülerdeki nesnelerin hareketli olup olmadığını belirleyecek parametredir. Bu parametre 3 nokta kombinasyonlarında maksimum yükseklik değerini belirler.

Çizelge 4.2.'de görülüyor ki, belirlenen ön tanımlı değerden daha büyük bir değeri alabilmeniz için yüksek hızlarda hareket eden cisimleri arıyor olmanız gerekir. Çünkü cisimler hızlı hareket ettiklerinde aldıkları mesafe artacaktır.

Çizelge 4.2. HEIGHT_MAX parametresi karşılaştırılması. HEIGHT_MAX parametresi [0.05-1.00] arasında 0.05 adım aralığında değişirken, diğer parametreler TRAVEL_MIN = 0.5 , TOLERANCE = 1.0, SNR_MIN = 10 olarak alınmıştır (Atay vd., 2016)

2000CA30						Astronomia					
HEIGHT_MAX (piksel)	AsPlot Asteroid Sayısı	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Çalışma Süresi (sn)	HEIGHT_MAX (piksel)	AsPlot Asteroid Sayısı	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Çalışma Süresi (sn)
0.05	7	6	0	4	69	0.05	6	2	0	0	24
0.10	7	7	0	6	67	0.10	6	2	0	0	24
0.15	7	7	1	11	66	0.15	6	2	0	0	24
0.20	7	7	1	12	70	0.20	6	2	0	0	24
0.25	7	7	2	16	70	0.25	6	2	0	0	24
0.30	7	7	2	17	78	0.30	6	2	0	2	24
0.35	7	7	2	19	76	0.35	6	2	0	2	24
0.40	7	7	3	21	79	0.40	6	2	0	2	24
0.45	7	7	4	27	78	0.45	6	2	0	3	24
0.50	7	7	4	28	78	0.50	6	2	0	3	24
0.55	7	7	4	28	66	0.55	6	2	0	3	24
0.60	7	7	4	28	68	0.60	6	3	0	3	24
0.65	7	7	4	29	65	0.65	6	3	0	3	24
0.70	7	7	6	27	67	0.70	6	3	0	3	25
0.75	7	7	6	28	67	0.75	6	3	0	3	24
0.80	7	7	6	28	70	0.80	6	3	0	3	24
0.85	7	7	7	29	70	0.85	6	3	0	3	24
0.90	7	7	8	28	68	0.90	6	3	0	3	21
0.95	7	7	9	29	66	0.95	6	3	0	3	24
1.00	7	7	9	29	66	1.00	6	3	0	3	24

(a) 2000CA30

(b) Astronomia

4.3.2. TRAVEL_MIN parametresi

Hareketli cismin hızına bağlı olarak gidebileceği en kısa mesafeyi belirleyen parametre ve ardışık iki görüntü arasında hareketli cismi yakalayabilmek için gereken minimum uzaklık (piksel biriminde) değeridir. Çizelge 4.3’de görüldüğü gibi bu değer yükseldikçe daha hızlı cisimleri bulabilirken yavaş cisimleri bulamayacaktır. Kullanılan veri setine göre ihtiyaç duyulan değerde yükseltmek kullanıcıya sadece çok küçük bir zaman kaybı oluşturmaktadır. Ancak bunun için veri setinin alanında bozulmalar olmaması gerekir. Aksi durumda parametre değerini yükseltmek sadece kodun çalışma süresini uzatacaktır.

Çizelge 4.3. TRAVEL_MIN parametresi karşılaştırılması. TRAVEL_MIN parametresi [0.1 -3.00] arasında 0.10 adım aralığında değişirken, diğer parametreler HEIGHT_MAX = 0.15, TOLERANCE = 1.0, SNR_MIN = 10 olarak alınmıştır (Atay vd., 2016)

2000CA30						Astronomia					
TRAVEL_MIN	AsPlot Asteroid Sayısı	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Çalışma Süresi (sn)	TRAVEL_MIN	AsPlot Asteroid Sayısı	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Çalışma Süresi (sn)
0.1	7	7	430	449	657	0.1	6	3	3	38	23
0.2	7	7	2	45	92	0.2	6	3	2	4	26
0.3	7	7	0	8	74	0.3	6	2	0	0	24
0.4	7	7	0	4	62	0.4	6	2	0	0	24
0.5	7	7	0	2	30	0.5	6	2	0	0	24
0.6	7	7	0	2	29	0.6	6	2	0	0	24
0.7	7	7	0	0	26	0.7	6	2	0	0	24
0.8	7	7	0	0	26	0.8	6	2	0	0	24
0.9	7	7	0	0	27	0.9	6	2	0	0	24
1.00	7	7	0	0	27	1.00	6	2	0	0	24
1.10	7	7	0	0	26	1.10	6	2	0	0	24
1.20	7	7	0	0	27	1.20	6	2	0	0	24
1.30	7	7	0	0	27	1.30	6	1	0	0	24
1.40	7	7	0	0	26	1.40	6	1	0	0	24
1.50	7	7	0	0	26	1.50	6	1	0	0	24
1.60	7	7	0	0	27	1.60	6	1	0	0	26
1.70	7	7	0	0	26	1.70	6	0	0	0	10
1.80	7	7	0	0	27	1.80	6	0	0	0	10
1.90	7	7	0	0	27	1.90	6	0	0	0	10
2.00	7	5	0	0	27	2.00	6	0	0	0	10
2.10	7	5	0	0	27	2.10	6	0	0	0	10
2.20	7	5	0	0	29	2.20	6	0	0	0	10
2.30	7	5	0	0	26	2.30	6	0	0	0	10
2.40	7	4	0	0	27	2.40	6	0	0	0	10
2.50	7	4	0	0	26	2.50	6	0	0	0	10
2.60	7	3	0	0	27	2.60	6	0	0	0	10
2.70	7	3	0	0	27	2.70	6	0	0	0	10
2.80	7	2	0	0	26	2.80	6	0	0	0	10
2.90	7	2	0	0	26	2.90	6	0	0	0	10
3.00	7	2	0	0	26	3.00	6	0	0	0	10

(a) 2000CA30

(b) Astronomia

4.3.3. SNR_MIN parametresi

SNR parametresi; gözlenen cismin parlaklığı, görüntünün poz süresi ve gecelik seeing gibi etkenlere ve kullanılan veri setinin ne kadar kirli veya temiz olduğuna, görüntü alınan CCD'nin kalitesine bağlıdır. SNR_MIN'nin düşük olması, daha çok sayıda zayıf noktaların ölçülebilmesine sebep olur. Yüksek SNR_MIN'de, zayıf noktalar gürültü seviyesinin içerisinde kalarak ayırt edilemezler ve ölçülmeleri giderek zorlaşır. Dolayısıyla Çizelge 4.4'de görüldüğü gibi bu parametrenin değeri yükseltildikçe kodun çalışma süresini azalmaktadır. Ancak parametre değeri arttırıldığında tespit edilen cisim sayısı da azalmaktadır.

Çizelge 4.4. SNR_MIN parametresi karşılaştırılması. SNR_MIN parametresi [2.5-40] arasında 2.5 adım aralığında değişirken, diğer parametreler HEIGHT_MAX = 0.15, TOLERANCE = 1.0, TRAVEL_MIN = 0.5 olarak alınmıştır (Atay vd., 2016)

2000CA30						Astronomia					
SNR_MIN	AsPlot Asteroid Sayısı	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Çalışma Süresi (sn)	SNR_MIN	AsPlot Asteroid Sayısı	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Çalışma Süresi (sn)
2.5	7	7	0	6	71	2.5	6	2	0	0	24
5	7	7	0	6	64	5	6	2	0	0	23
7.5	7	7	0	6	71	7.5	6	2	0	0	24
10	7	7	0	6	64	10	6	2	0	0	24
12.5	7	6	0	6	71	12.5	3	2	0	0	24
15	7	5	0	6	62	15	6	2	0	0	24
17.5	7	5	0	6	69	17.5	6	2	0	0	23
20	7	4	0	6	30	20	6	2	0	0	23
22.5	7	3	0	6	25	22.5	6	2	0	0	23
25	7	3	0	5	25	25	6	2	0	0	23
27.5	7	3	0	5	23	27.5	6	2	0	0	23
30	7	2	0	6	23	30	6	2	0	0	27
32.5	7	2	0	6	22	32.5	6	2	0	0	25
35	7	2	0	6	21	35	6	2	0	0	26
37.5	7	2	0	5	20	37.5	6	2	0	0	25
40	7	2	0	5	23	40	6	2	0	0	25

(a) 2000CA30

(b) Astronomia

4.3.4. TOLERANCE parametresi

TOLERANCE parametresi, zamana bağlı olarak bir hareketli cismin birim zamanda gidebileceği mesafe miktarını göstermektedir. İlk iki noktanın üçüncü nokta ile bir doğru oluşturup oluşturmayacağını belirleyecek olan önemli bir parametredir. Bu parametrenin algoritmadaki önemi Bölüm 3.2'te anlatıldı. Çizelge 4.5'de görüldüğü gibi cismin gerçekten hareketli olup olmadığını belirleyeceği için bu değer gereğinden fazla büyütülürse daha çok belirsiz nesne tespit edilmesine neden olacaktır.

Çizelge 4.5. TOLERANCE parametresi karşılaştırılması. TOLERANCE parametresi [0.25-4.00] arasında 0.25 adım aralığında değişirken, diğer parametreler HEIGHT_MAX = 0.15, TRAVEL_MIN = 0.5, SNR_MIN = 10 olarak alınmıştır. (Atay vd., 2016)

2000CA30						Astronomia					
TOLERANCE (piksel)	AsPlot Asteroid Sayısı	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Çalışma Süresi (sn)	TOLERANCE (piksel)	AsPlot Asteroid Sayısı	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Çalışma Süresi (sn)
0.25	7	7	0	2	73	0.25	6	2	0	0	25
0.50	7	7	0	4	71	0.50	6	2	0	0	24
0.75	7	7	0	6	72	0.75	6	2	0	0	24
1.00	7	7	0	6	71	1.00	6	2	0	0	24
1.25	7	7	1	8	73	1.25	6	2	0	0	24
1.50	7	7	1	9	73	1.50	6	2	0	0	24
1.75	7	7	1	11	70	1.75	6	2	0	0	24
2.00	7	7	2	11	74	2.00	6	2	0	0	24
2.25	7	7	2	13	75	2.25	6	2	0	0	25
2.50	7	7	2	13	72	2.50	6	2	0	0	25
2.75	7	7	2	15	74	2.75	6	2	0	0	24
3.00	7	7	2	14	74	3.00	6	2	0	0	24
3.25	7	7	3	14	72	3.25	6	2	0	0	24
3.50	7	7	4	12	75	3.50	6	2	0	0	25
3.75	7	7	4	13	72	3.75	6	2	0	0	24
4.00	7	7	4	14	75	4.00	6	2	0	0	24

(a) 2000CA30

(b) Astronomia

4.4. Farklı Verilerle A-Track Performans Testi

Tübitak Ulusal Gözlem Evi'nden T100 teleskobu ile 2012 yılında farklı mevsimlerde alınmış veriler ile A-Track programının testi yapıldı. Her bir veri seti ilgili projede hedeflenen asteroidlere aittir. Bu performans testinde kullanılabilecek temiz ve düzgün veriler seçildi. Çizelge 4.6'da görülen veriler, çok sayıda veriler arasından tercih edilen verilere aittir.

Çizelge 4.6. Test edilen 11 veri setine ait sonuçlar

Veri Adları	Cisim Adları	A-Track Doğru	A-Track Yanlış	Belirsiz Nesne	Time
20120620	32116_2	1	0	3/0	00:03:15
20120621	1999JU_1	2	0	1/0	00:00:29
20120701	5326	1	0	0	00:01:14
20120805	B_01	2	0	0	00:00:38
	B_02	5	0	0	00:01:24
20121016	B	2	0	1/0	00:01:49
20121017	B_B	1	0	0	00:00:18

20121018	A_03	3	1	1/0	00:02:59
	B_03	1	0	0	00:01:01
20121019	B_01	3	0	2/0	00:02:56
20121020	GRB12042 2	1	0	0	00:03:22

Çizelge 4.6’da ki değerler; üçüncü sütun tespit edilen asteroidlerin kaç tanesinin doğru olduğunu, dördüncü sütun tespit edilen asteroidlerin kaçının hatalı olduğunu göstermektedir. Beşinci sütun ise belirsiz nesne sayısının kaçının doğru olduğunu, altıncı sütun da belirsiz obje sayının kaçının yanlış tespit olduğunu göstermektedir. Bu çizelgede kullanılan verilerden de görülüyor ki, programın uzun süreler çalıştığı veri setlerinde uzun çalışma süresi dışında olumsuz bir etki olmamaktadır.

5. SONUÇ

Bu tez çalışmasında, DTY (Chen T., Chung K., 2001) güneş sistemi cisimlerinin tespiti için geliştirilerek Multiple Image Line Detection (MILD) (Çoklu Görüntüden Doğru Tanıma Algoritması) adı verilen algoritma oluşturulmuştur. Çalışma sırasında tek bir görüntüde doğru tanıma amacıyla kullanılan kriterler ardışık görüntülere uygulanabilir hale getirilmiş, gerekli kriterler üretilmiştir. Algoritmanın uygulanması sırasında aday hareketli cisimlerin hızları da kontrol kriterlerine eklenmiştir. Fazla sayıda teleskop görüntüsüne ihtiyaç duymayan, belli bir teleskop sistemine bağımlı olmayan, görsel doğrulamaya gerek bırakmayan ve şimdiye kadar bu alanda hiç uygulanmamış bir yöntemle program geliştirilmiştir.

Geliştirilen MILD algoritması Python3X programlama dili kullanılarak kodlanmış ve açık kaynak kodlu, modüler, geliştirmeye uygun bir program olan A-Track yazılımı üretilmiştir (Bknz. EK-1). Yazılım aynı geceye ve hemen hemen aynı bölgeye ait ardışık 3 ila 15 FITS dosyası ile çalışmakta ve hareketli cisim adayı olarak tespit ettiği cisimleri raporlamaktadır. Ek olarak, yazılım, ürettiği bir hareketli-gif dosyası içerisinde tespit ettiği hareketli cisimleri işaretlemekte ve aday cisimleri kullanıcıya animasyon olarak da sunmaktadır. Bu özellikleri ile gecelik gözlemlerin hızlı bir şekilde hareketli cisim taramasından geçirilebilmesini olanaklı kılmaktadır.

Yazılımın gereksinim duyduğu, ön tanımlı olarak belirlenmesi gereken parametreler (TRAVEL_MIN, HEIGHT_MAX, SNR_MIN vb.) konfigürasyon dosyasında belirtilmiştir. Bu parametreler için verilen ön tanımlı değerleri elde etmek için farklı veri setleri parametrelerin kullanıma uygun aralıkları için tekrar tekrar test edilmiştir. Sonuçlar tez içerisinde Bölüm 5'de paylaşılmıştır. Bulunması hedeflenen hareketli cisimlerin hızlı veya yavaş, parlak veya sönük olması durumunda bu parametre değerlerini konfigürasyon dosyasında uygun şekilde değiştirmek son kullanıcıya bırakılmıştır.

A-Track yazılımı bu alanda çok kullanılan Astrometrica ve son dönemde geliştirilmiş Collitec yazılımları ile de karşılaştırmalı testlere tabi tutulmuştur. Yapılan performans testlerinde bu yazılımlardan çok daha kısa çalışma sürelerinde en az bu programlar kadar hareketli cisim tespiti yapılabildiği görülmüştür. Yapılan performans testleri ile ilgili sonuçlar Bölüm 6'da verilmiştir.

İlerleyen zamanlarda, A-Track programı eklenecek modüllerle geliştirmeye uygundur. Bu sayede örneğin tespit edilen asteroidlerin astrometrisi yapılabilir ve yörüngesi çıkartılan cisimlerin yörünge parametreleri elde edilebilir.

A-Track yazılımı pipeline kullanıma uygunluğu ile eski verilerin hareketli cisim taramasından geçirilmesine de uygundur. Bu çalışmada 2012-2016 yılları arasındaki 100 civarında veri bu yaklaşımla işlenmiştir. Ayrıca, yazılım komut satırından

çalıştırmaya uygun olduğu gibi, son kullanıcılara pratik bir kullanım sunmak amacıyla, bir de arayüze sahiptir. Arayüz kullanımı sayesinde yazılımın amatör astronomlar açısından da ilgi görmesi beklenmektedir (Bknz. EK-2).

Yazılımın önemli bir yanı da, paralel programlama kütüphanelerinin kullanılmasıdır. Bu sayede, yazılımın en yüksek CPU zamanı tüketimine sahip olan modüllerinin kullanılan bilgisayarın çekirdek sayısı ile orantılı olarak hızlı çalışmaktadır. Çalışma sonucunda, geliştirilen MILD algoritması ile tasarlanan A-Track programı ve Python-GUI araçları kullanarak bir arayüz tasarlandı. Şu an Ubuntu, Fedora ve Mac işletim sistemlerinde çalışabilir durumdadır.

6. KAYNAKLAR

ANONİM, 2016. <https://neutrium.net/mathematics/basics-of-affine-transformation/> [Son erişim tarihi: 20.10.2016].

ANONİM, 2016. <https://neutrium.net/mathematics/basics-of-affine-transformation/> [Son erişim tarihi: 06.11.2016].

ANONİM, 2017. http://fits.gsfc.nasa.gov/fits_standard.html [Son erişim tarihi: 06.11.2016].

ANONİM, 2017. <http://www.starlink.rl.ac.uk/docs/sun226.htx/mud165.pdf> [Son erişim tarihi: 12.03.2017].

ATAY T. ve ark.2016. A-Track: A new approach for detection of moving objects in FITS images. *Computer Physics Communications*, 207 ,524–530

BERTIN, E. and ARNOUTS, S.1996."SExtractor: Software For Source Extraction". *Astronomy and Astrophysics Supplement Series*, 117, 393-404.

CAIN S.C. ve MACDONALD A. 2007. *Aerospace Conference*, IEEE.

CINSINDIKCI M. 1997. Tez- 01/09/97. MSc., “Neural Network Solutions for ATM Routing & Multicasting Problems.”, Ege Univ., Computer Eng. Dep.,

CHAMBERLIN A. B. Near-Earth Asteroid Discoveries, . [Son erişim tarihi: 01.07.2017] <https://cneos.jpl.nasa.gov/stats/>

CHEN T. and CHUNG K. 2001. "A New Randomized Algorithm For Detecting Lines". *Real-Time Imaging*, 7, 473-481.

YANAGISAWA T. ve NAKAIMA A. 2007. *Publ. Astron. Soc. Japan*, 57, 399–408.

GUIDO VAN ROSSUM vd. *Style Guide for Python Code*. [Son erişim tarihi: 01.07.2017] <http://legacy.python.org/dev/peps/pep-0008/>

HOUGH, P.V.C. 1962. "Method And Means For Recognizing Complex Patterns". *U.S. Patent* 3, 069, 654.

KUBICA J. vd. 2007. *Icarus* 189, 151–168.

LANG D. vd. 2010 Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal*, 139(5):1782-800.

MOHANTY, N. C. 1981. *IEEE Trans. Pattern Anal. Machine Intell.*, 3, 606.

O’CONNOR A. C. vd. 2012. *Aerospace and Electronics Conference (NAECON)*. IEEE National.

- PENCE, W. D. 2010. De_nition of the exible image transport system (_ts), version 3.0. *Astronomy and Astrophysics*, 524:A42.
- PETER S. G. vd. 2005. *The Astronomical Journal*, 130:1951–1960.
- RAAB H. 2012. Astrometrica: Astrometric data reduction of CCD images. *Astrophysics Source Code Library*.
- GONZALEZ R.C. , WOODS R.E. 2008. Digital Image Processing Third Edition. *Pearson Prentice Hall*.
- SAVANEVICH, V. E. and KOZHUKHOV A.M. 2012. “Program of Automated Asteroids Detection CoLiTec --- New Features and Results of Implementation”. *43rd Lunar and Planetary Science Conference*.
- TEWES M. 2015. A python package to quickly, automatically, and robustly identify geometrical transforms between optical astronomical images.
- ÜNSAL F. 2009. TMMOB Harita ve Kadastro Mühendisleri Odası. *12.Türkiye Harita Bilimsel ve Teknik Kurultayı*.

7. EKLER

Ek-1: A-Track Programı Kurulumu ve Kullanımı

Kolay okunabilir olması (PEP8 standardında (Guido van Rossum, 2015)), platformdan bağımsız ve astronomi alanında bir çok modüle sahip olması bakımından Python 3.x dili tercih edildi. MILD yazılımın testleri GNU/Linux (Fedora 20, Ubuntu 14.10, vb.) ve Mac OS X (10.10.4, Yosemite) üzerinde başarıyla çalıştırıldı. Windows'ta A-Track'i kullanmak istenildiğinde öncelikle SExtractor yüklenmelidir. Bu yükleme sorunlar içerdiğinden test aşamasında tercih edilmedi.

Kurulum için <https://github.com/akdeniz-uzay/A-Track> linkinde bahsedildiği gibi, A-Track'i Linux veya Mac'e kurmak için A-Track paketini indirip install_linux.sh (Linux için: sudo sh install_linux.sh) veya install_mac.sh (Mac için: sh install_mac.sh) yükleme komut dosyalarını çalıştırmanız yeterlidir.

Alternatif olarak, aşağıdaki adımları izlenerek A-Track manuel olarak kurulabilir:

- 1- Python3, pip3, imagemagick, git ve SExtractor'ı yükleyin:
Ubuntu:
Sudo apt-get install python3 python3-dev python3-pip imagemagick sextractor libxt-dev git build-essential
- 2- Fedora:
Sudo dnf install python3 python3-devel python3-pip imagemagick sextractor libXt-devel git make automake gcc gcc-c ++ çekirdek-devel
En son SExtractor'ı buradan yükleyin (eski sürümler daha az nesneyi tespit ettiği için v2.19.5 öneriliyor).
- 3- Mac OS X:
Brew install python3 python3-pip imagemagick git-all sextractor
(Bağımlılıkları yüklemek için Homebrew'a ihtiyaç duyuluyor.)
- 4- Numpy, Pandas, Scipy, pyFITS ve yastıkı pip3 kullanarak kurun:
Sudo pip3 install scipy pandas numpy pyfits yastık
(Mac kullanıcıları sudo kullanmazlar.)
- 5- Astrosciidata'yı indirin ve yükleyin:
Git clone <https://github.com/japs/astrosciidata>.git
Cd astrosciidata
Sudo python3 setup.py install
(Mac kullanıcıları sudo kullanmazlar.)
- 6- Alipy'yi indirin ve yükleyin:
Git clone <https://github.com/akdeniz-uzay/alipy>.git
Cd alipy
Sudo python3 setup.py install

(Mac kullanıcıları sudo kullanmazlar.)

7- A-Track paketini indirin ve f2n'yi kurun:

Git clone <https://github.com/akdeniz-uzay/A-Track.git>

Cd A-Track / f2n

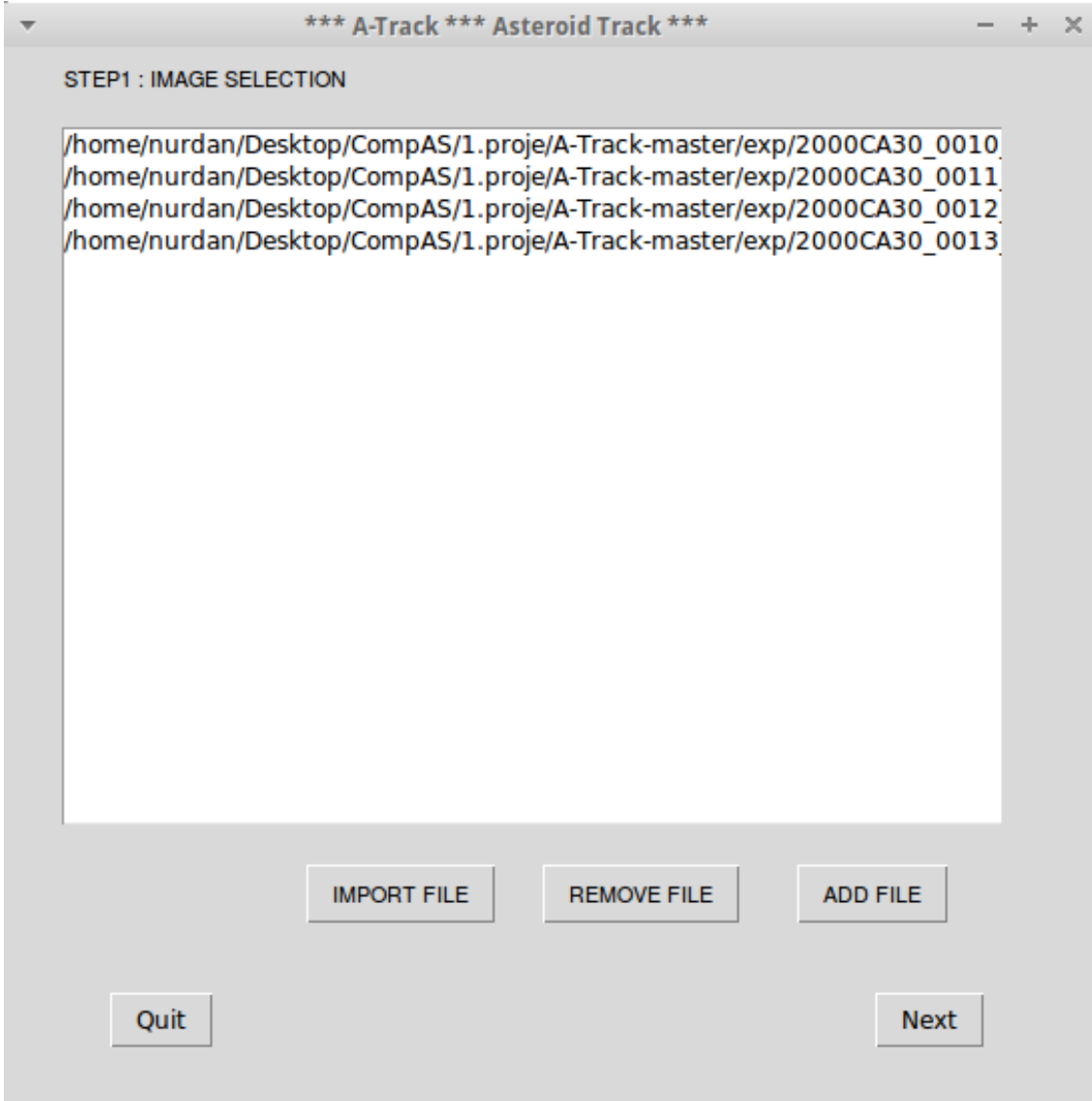
Sudo python3 setup.py install

(Mac kullanıcıları sudo kullanmazlar.)

Şimdi, A-Track dizininde bir komut satırı arabirimi açılıp A-Track çalıştırılabilir.

Ek-2: A-Track Programının Arayüzü

A-Track programı tasarlanırken temel amaç kullanıcı dostu, kolay erişilebilir, kurulabilir ve kullanılabilir olmasıydı. Bu amaca ulaşmak için Python 3.5 sürümü ile otomatik olarak gelen Tkinter modülden yararlanarak bir arayüz tasarlandı. Bu arayüze Github sayfasından <https://github.com/nurdankarapinar/A-Track-GUI> linki ile ulaşılabilir.



Şekil 7.1. A-Track arayüzü 1.sayfa görüntüsü

atrack.py dosyası ile aynı dizinde bulunacak olan mainprogram.py çalıştırıldığında ilk önce görüntülerin yükleneceği giriş sayfası olan birinci açılacaktır. Kullanıcı bu dosyayı çalıştırdığında arka planda A-track.py dosyanın bulunduğu dizinde bir klasör oluşacaktır. Görüntüleri seçtikten sonra 'NEXT' butonu ile ilerlendiğinde seçilen görüntüler oluşturulan klasörün içerisine kopyalanacaktır (Şekil 7.1).

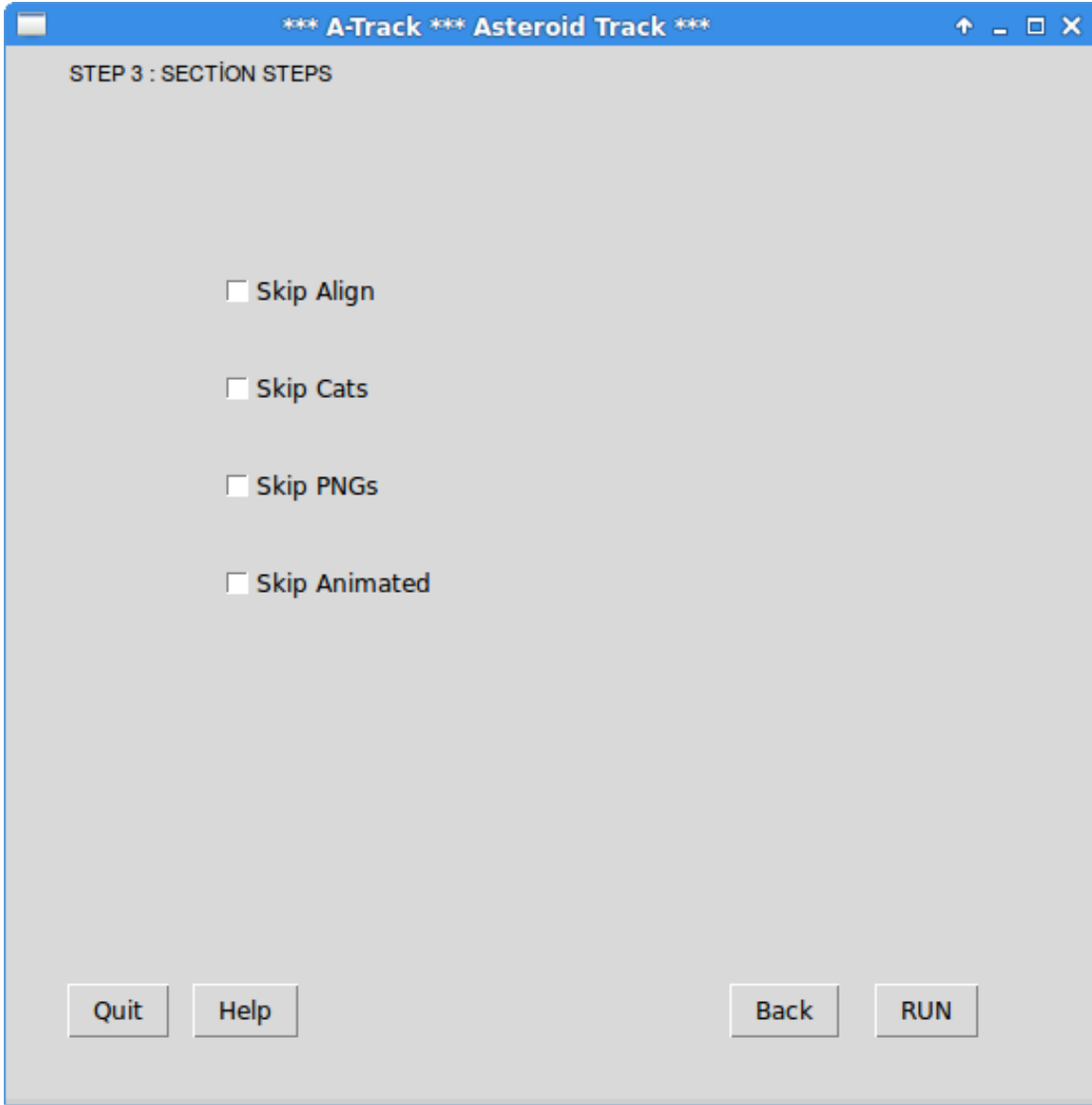
***** A-Track *** Asteroid Track *****

STEP 2 : PARAMETERS

DETECT_THRESH	3	1-3 (sigma)
ANALYSIS_THRESH	3	integer (n ≤ 32)
DETECT_MINAREA	1	3-50
PIXEL_SCALE	0.31	arcsec
SEEING_FWHM	1.5	arcsec
PHOT_AUTOPARAMS	"2.5, 3.5"	min_radius
BACK_SIZE	64	<width>,<height>
BACK_FILTERSIZE	3	<width>,<height>
DEBLEND_NTHRESH	16	arcsec
SATUR_LEVEL	60000	ADU
DEBLEND_MINCONT	0.00001	arcsec
GAIN	0.55	e-/ADU
rerun	True	True or False
keepcat	True	True or False
verbose	False	True or False
FWHM_MIN	1	float
FLUX_MAX	500000	integer
ELONGATION_MAX	1.8	integer
SNR_MIN	10	integer
TRAVEL_MIN	0.5	pixelscale
HEIGHT_MAX	0.1	floats
SCALE	0.31	floats
V_MAX	0.03	px/min
TOLERANCE	1.0	floats
SPEED_MIN	0.1	px/min

Şekil 7.2. A-Track arayüzü 2.sayfa görüntüsü

İkinci sayfada temelde deneysel olarak bizim belirlenen .config dosyasındaki parametreler ve değerleri görünür. Bu sayfadaki değerler, herhangi bir teleskop sistemine bağlı olmadan çalışabilen A-Track programının değiştirilebilir değerleridir. İstenilirse bu değerler ile optimum sonuçlar elde edilebilir. Veya kendi görüntülerinizin değerlerine göre düzenledikten sonra 'NEXT' butonu ile ilerlediğinizde, görüntülerin kopyalandığı dizine user.config olarak kaydedilecektir (Şekil 7.2).



Şekil 7.3. A-Track arayüzü 3.sayfa görüntüsü

Üçüncü sayfada A-Track programının çalışma aşamaları kullanıcının isteğine bağlı olarak seçilebilir buton şeklindedir. Hiçbir butonu işaretlemezseniz A-Track programı tüm aşamaları ile çalışacaktır. İşlem süresinin daha kısa sürmesi ve yapılması istenilmeyen aşamalar varsa işaretlenen butonlardaki aşamalar yapılmayacaktır. Seçim işlemi tamamlandıktan sonra 'RUN' butonu ile ilerlendiğinde atrack.py dosyası çalışacaktır (Şekil 7.3).

*** A-Track *** Asteroid Track ***

STEP 4 : RESULT FILE

ObjectID	FileID	Flags	x	y	Flux	Background	Speed(px/min)
1	0	3	1690.9832	598.4767	98357.80	18042.16	0.153932
1	1	3	1690.7949	596.0910	98185.68	18163.78	0.153932
1	2	3	1690.3448	593.8844	86647.27	19137.78	0.153932
1	3	0	1690.2729	591.9268	63879.91	28560.11	0.153932
2	0	0	1325.1971	812.6868	32650.53	18391.83	0.145607
2	1	0	1327.4049	812.6995	31262.20	18485.28	0.145607
2	2	0	1329.4390	812.7189	27705.92	19458.97	0.145607
3	0	0	743.7114	878.6750	45239.05	18035.42	0.234854
3	1	0	746.2470	876.1613	55407.46	18135.04	0.234854
3	2	0	748.8260	873.7244	68127.62	19124.41	0.234854
3	3	0	751.0767	871.8347	51408.53	28798.86	0.234854
5	0	0	518.3959	1714.3356	45763.70	17235.51	0.182293
5	1	0	520.9368	1715.0139	52509.91	17290.53	0.182293
5	2	0	523.4965	1715.7223	50745.41	18209.05	0.182293
5	3	0	525.9130	1716.4252	36693.74	27511.20	0.182293
6	0	0	596.0693	1594.9628	92663.20	17523.44	0.170291
6	1	0	598.3350	1593.7904	90798.25	17583.01	0.170291
6	2	0	600.5266	1592.5853	85324.80	18529.65	0.170291
6	3	0	602.5336	1591.5961	59567.09	27948.40	0.170291
8	0	0	1767.6322	1329.9083	28127.78	18295.90	0.196407
8	1	0	1769.8590	1331.9574	33088.01	18354.79	0.196407

Quit Help Go To Result

Şekil 7.4. A-Track arayüzü 4.sayfa görüntüsü

Son sayfada işlem sonucu ekrana gelecek. Bu sayfada, 'EXIT' butonu ile çıkış yapılabilir veya görselleştirme işlemlerini seçerek öngörüm yapılabilir. Esasen son sayfada gördüğünüz sonuçlar görüntüler ile aynı dizin altında result.txt dosyası olarak kullanıcıya verilir (Şekil 7.4).

ÖZGEÇMİŞ



Nurdan KARAPINAR 1989 yılında İzmir'de doğdu. İlköğrenimine Batman'da başlayıp, orta ve lise öğrenimini Uşak'ta tamamladı. 2009 yılında girdiği Ege Üniversitesi Fen Fakültesi Astronomi ve Uzay Bilimleri Bölümü'nden 2014 yılında Astronom olarak mezun oldu. Eylül 2014 - Temmuz 2017 yılları arasında, Akdeniz Üniversitesi Fen Bilimleri Enstitüsü, Uzay Bilimleri ve Teknolojileri Anabilim Dalı'nda Yüksek Lisans öğrenimini tamamladı.