
TFG: Detección semi-automática de Sistemas de Estrellas Dobles



Trabajo de Fin de Grado

David Antuña Rodríguez
Daniel Gutiérrez Gertrúdix
Javier Carrión García

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Curso 2017/2018

TFG: Detección semi-automática de Sistemas de Estrellas Dobles

Trabajo de Fin de Grado

Dirigida por el Doctor

Rafael Caballero Roldán

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

Curso 2017/2018

Agradecimientos

Queremos dar las gracias a nuestro director, Rafael Caballero, por el esfuerzo y la dedicación a este trabajo, y por habernos aguantado durante todo el año. Ha sido un placer haber tenido la oportunidad de trabajar no solo con un gran científico sino con una persona maravillosa.

Resumen

La detección de sistemas binarios tiene gran importancia en la astrofísica, pues permite, por ejemplo, calcular con precisión la distancia a la que se encuentran o descubrir nubes de materia oscura mediante la detección de cambios inesperados en las órbitas. Sin embargo, la mera ocurrencia de dos estrellas cercanas en el cielo no supone que se trate de un sistema binario, porque pueden verse así por efecto de la perspectiva. Para saber con seguridad que en efecto es un sistema binario habría que observar cómo giran una alrededor de la otra, pero esto a menudo no es posible al tratarse de órbitas que requieren decenas de miles de años. Un mecanismo indirecto para detectar nuevas estrellas binarias es descubrir dobles con un notable movimiento propio común. La idea es que es estadísticamente improbable encontrar parejas que se muevan muy rápido, y en la misma dirección y sentido.

Este trabajo pretende colaborar en la detección de estos pares con alto movimiento propio común como posibles “candidatas” a estrella binarias. Para ello utilizamos la superposición de imágenes tomadas por telescopios profesionales con una diferencia temporal cercana a los 50 años. La detección se realiza comprobando parejas cuya posición varía de forma significativa entre ambas imágenes. Para ello hemos realizado un programa que analiza de forma automática estas imágenes y sugiere posibles candidatas que deben ser corroboradas por el usuario. El sistema descarta el 97 % de las imágenes que no tienen estrellas dobles, y encuentra el 33 % de las imágenes que sí la tienen.

Índice

Agradecimientos	v
Resumen	vii
1. Introducción	1
1.1. Estrellas dobles	1
1.2. Problema	2
1.3. Importancia	2
1.4. Nuestra propuesta	2
2. Introduction	3
2.1. Double stars	3
2.2. Issue	4
2.3. Importance	4
2.4. Our approach	4
3. Estrellas dobles	5
3.1. Centroides	5
3.2. Conceptos	5
Notas bibliográficas	5
4. Recolección de datos	7
4.1. Aladin	7
4.2. Obtención de imagenes	7
4.3. WDS	8
5. Workflow	11
5.1. Introducción	11
5.2. Estructura	11
5.3. Etapas	12
5.4. Uso	13

6. Procesamiento	15
6.1. Recoloreado	15
6.2. Comprobación	16
6.3. Contador de pixels	18
6.4. Corte	19
7. Parametrización del detector	23
7.1. Introducción	23
Notas bibliográficas	23
8. Problemas	25
8.1. Introducción	25
Notas bibliográficas	25
Bibliografía	27

Índice de figuras

3.1. Estrella ideal vs Estrella real	6
4.1. Imagenes descargadas de las bases astronómicas	8
4.2. Superposición de las imagenes	8
6.1. Recoloreado usando la distancia Manhattan	16
6.2. Recoloreado de la imagen 6.1a con un umbral de 150	17
6.3. Imágen sin recortar	20
6.4. Imagenes resultantes de la división	21

Índice de Tablas

Capítulo 1

Introducción

RESUMEN: En primer lugar vamos a aclarar qué es una estrella doble y cuál es su utilidad para la comunidad científica. También plantearemos el problema que este trabajo pretende resolver.

1.1. Estrellas dobles

A lo largo de la historia los astrónomos se han ocupado de las características de los diferentes objetos que se encuentran en el espacio: estrellas, planetas, galaxias, etc. Entre estos objetos, desde antiguo han llamado la atención las estrellas dobles.

Un pionero en este campo fue **Willen J. Luyten** quien se dedicó al estudio de estrellas con movimiento propio, y descubrió gran cantidad de estrellas múltiples sentando así las bases para el estudio de las mismas. (Luyten, 1971)

Según <http://www.astromia.com/glosario/binaria.html>

“Una Estrella doble o Estrella binaria, es una pareja de estrellas que se mantienen unidas por gravitación y giran en torno a su centro de masas común. Los periodos orbitales, que van desde minutos en el caso de dobles muy cercanas hasta miles de años en el caso de parejas distantes, dependen de la separación entre las estrellas y de sus respectivas masas. La observación de las órbitas de estrellas dobles es el único método directo que tienen los astrónomos para pesar las estrellas. En el caso de parejas muy próximas, su atracción gravitatoria puede distorsionar la forma de las estrellas, y es posible que fluya gas de una estrella a otra en un proceso denominado transferencia de masa.”

Aunque la mayoría de las estrellas que vemos son dobles o incluso múltiples, muy pocas de ellas son detectables a través de un telescopio debido a la gran cercanía de sus componentes. A estas se les llama dobles visuales, y es a las que nos dedicamos en este trabajo.

1.2. Problema

El reconocimiento de estrellas binarias es un trabajo muy dedicado porque se realiza de forma manual, es necesario analizar las fotografías una a una hasta que se encuentra un par de estrellas con unas características similares que encajen en el arquetipo de un sistema de estrellas dobles.

En este trabajo nos encargaremos del marco de las estrellas con un movimiento propio común elevado, es decir, parejas de estrellas que tengan un movimiento muy rápido, en el cual las componentes, dirección y sentido, coincidan.

1.3. Importancia

Esta tipología de estrellas alberga una importancia clave a la hora de recolectar datos tales como el peso de una estrella, como se menciona en la sección 1.1, o en caso de ya conocer las masas su distancia exacta.

Otra ventaja de conocer estas estrellas radica en el descubrimiento de nubes de materia oscura, dado que estos sistemas de estrellas se mueven en paralelo es posible observar cambios una de las estrellas que componen el sistemas que indiquen la presencia de cuerpos invisibles a simple vista como dichas nubes.

Hartkopf, Harmanec y Guinan (2007)

1.4. Nuestra propuesta

Queremos crear un sistema de detección de estrellas binarias con un alto movimiento propio común y, por tanto, observables en un periodo relativamente corto de tiempo, 50 años aproximadamente. Este proceso no sería completamente automático sino que una vez detectado una posible estrella doble tendría que ser revisada manualmente para cerciorarse de que no se trata de un falso positivo.

Capítulo 2

Introduction

ABSTRACT: First we will see what is considered a double star and the relevance it has for the scientific community. We will also raise the problem this work intends to solve.

2.1. Double stars

Throughout history astronomers have dealt with the features of different objects in space: stars, planets, galaxies, etc. Among them, attention has been drawn to double stars since ancient times.

A pioneer in this field was **Willen J. Luyten** who dedicated himself to the study of stars with common proper motion, and discovered a large number of multiple stars, laying the foundations for future investigation. (Luyten, 1971)

According to <http://www.astromia.com/glosario/binaria.html>.

“A Double Star or Binary Star is a pair of stars held together by gravity spinning around the common center of masses. The orbital periods, that oscillate between minutes for very close doubles to millions of years for the distant ones, depend on the separation between stars and their masses. The observation of the orbits of double stars is the only direct method the astronomers have to weight stars. For near pairs the gravitational pull may distort the shape of the stars, and gas may flow from one star to another in a process called mass transfer.”

Even though most of the stars we see are double or even multiple, very few of them are measurable using a telescope due to the closeness of its components. These are called visual doubles, and that is what we are dedicated to in this work.

2.2. Issue

The detection of binary stars is a very dedicated work because it is done manually, the pictures have to be analyzed one by one looking for a pair of stars with similar characteristics that fit the archetype of a double star system.

This work focuses in the frame of double stars with high proper motion, that is, pairs of stars with very fast movement in which the components, sense and direction, coincide.

2.3. Importance

This kind of stars have a key value to collect data such as the weight of a star, as stated in the section 2.1, or, knowing the masses, the exact distance.

Another benefit of these stars lies in the discovery of dark matter clouds, these star systems movement is parallel so it is possible to notice changes in the expected behaviour suggesting the presence of invisible masses such as said clouds.

Hartkopf, Harmanec y Guinan (2007)

2.4. Our approach

We want to create a system capable of detecting binary stars with high proper motion and, therefore, observable in a relatively short period of time, approximately 50 years. This process wouldn't be completely automatic, once a possible double star is detected it would have to be manually checked to make sure it is not a false positive.

Capítulo 3

Estrellas dobles

RESUMEN: Antes de abordar el problema que nos ocupa en este trabajo vamos a explicar los conceptos relativos a estrellas dobles que en el se tratan.

3.1. Centroides

Un punto fundamental para la detección de estrellas dobles consiste en la búsqueda y análisis de los centros de las estrellas, los llamados centroides. Esto es debido a que la estrella no aparece en las imágenes como un círculo, cuyo centro puede determinarse sin más (figura 3.1a), sino como una acumulación de puntos brillantes, con un contorno difuso (figura 3.1b).

3.2. Conceptos

Notas bibliográficas

...

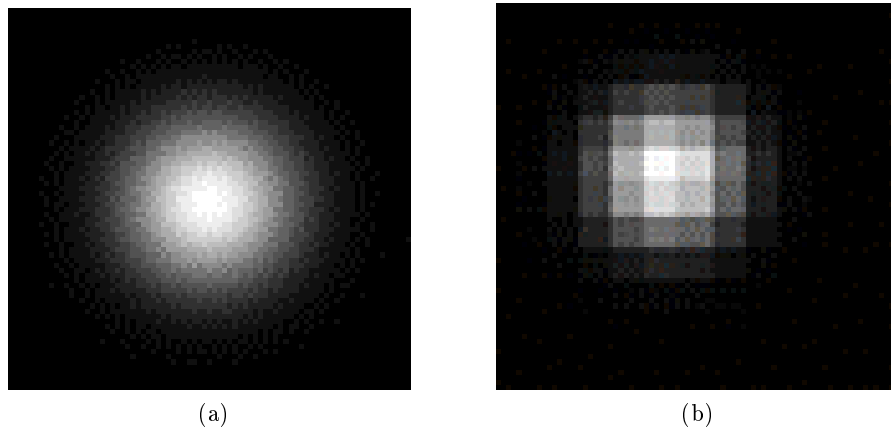


Figura 3.1: Estrella ideal vs Estrella real

Capítulo 4

Recolección de datos

RESUMEN: A lo largo de este capítulo veremos que herramientas hemos empleado para obtener imágenes astronómicas con las que trabajar, así como datos de estrellas dobles ya catalogadas.

4.1. Aladin

<http://aladin.u-strasbg.fr/>

Aladin Sky Atlas es un programa interactivo desarrollado por el **CDS**, Centro de Datos Astronómicos de Estrasburgo, que permite acceder recursos astronómicos digitalizados.

Esta herramienta no solo permite visualizar imágenes de cuerpos celestes a lo largo de los años, también permite realizar una superposición de las mismas para facilitar su comparación.

4.2. Obtención de imágenes

Como se describe la sección 4.1 es posible obtener imágenes que muestren la evolución de los cuerpos celestes en coordenadas concretas con años de diferencia.

Como se puede apreciar, figura 4.2, Aladin no solo superpone las imágenes, también las rota para tratar de encajarlas y colorea cada imagen de un tono de modo que las estrellas pueden ser rojas o azules, dependiendo de la imagen en que aparezcan, si el cuerpo coincide, no se ha movido, será blanca.

Para facilitar el proceso de descarga aladin permite emplear scripts en combinación con ficheros de parámetros. Puesto que no existe ninguna fuente que tenga fotos de todas las coordenadas celestes son necesarios dos scripts que emplean las siguientes fuentes.

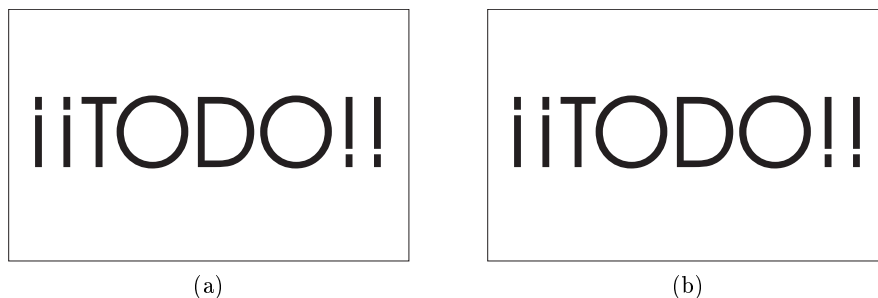


Figura 4.1: Imágenes descargadas de las bases astronómicas



Figura 4.2: Superposición de las imágenes

- **POSSI** y **POSSII** para descargar imágenes de coordenadas positivas.
- **POSSI** y **SERC** para las negativas.

Además de ejecutar la superposición el script hace zoom en la coordenada proporcionada como parámetro y almacena la imagen resultante.

4.3. WDS

`http://ad.usno.navy.mil/wds/`

La obtención de recursos de trabajo ya ha sido resuelta en la sección 4.2, sin embargo, es necesario obtener un listado de sistemas de estrellas dobles ya reconocidas que poder analizar con el fin de semi-automatizar su reconocimiento.

WDS, Washington Double Star Catalog, es un catálogo mantenido por el Observatorio Naval de los Estados Unidos que recopila información sobre sistemas de estrellas múltiples.

El problema es que el catálogo almacena gran cantidad de estrellas, y nosotros queremos casos de sistemas de estrellas dobles de movimiento rápi-

do. Con el fin de obtener la información que nos interesa hemos creado un programa que filtra dichos datos basandose en los siguientes criterios:

- La última vez que se vió el sistema debe ser como mínimo 1975, esto asegura que en los catalogos existiran almenos dos imagenes, una relativamente actual y una de los años 50.
- La magnitud de las estrellas que componen el sistema deben ser a lo sumo 19, de este modo las estrellas seran apreciables a simple vista.
- La separación debe pertenecer al rango entre 2 y 180, las estrellas con valores mayores están demasiado alejadas y desvirtuarían los resultados.
- El desplazamiento ha de ser superior a 60 para que el movimiento sea apreciable.

Las coordenadas de todos los sistemas que sigan dichos criterios serán almacenadas en un fichero que puede usarse como parametro de los scripts descritos en la seccion 4.2.

Capítulo 5

Workflow

RESUMEN: Una vez resuelta la recolección de datos es necesario ver que tratamientos aplicarles para alcanzar nuestro objetivo, pero antes vamos a definir una plataforma genérica sobre la cuál desarrollar dichos procesos.

5.1. Introducción

En los capitulos 6 y 7 se describen diferentes procesos por los que una fotografía ha de pasar para poder detectar sistemas de estrellas dobles, cada una de estas etapas viene definida por un número arbitrario de entradas y salidas y un evento que al abortar la ejecución permita al programa salir de forma segura.

Cada etapa por tanto puede ser configurada y lanzada por separado puesto que para conectarlas solo es necesario que la salida de una etapa sea la entrada de la siguiente, de esta forma todas las etapas pueden trabajar de forma simultánea. Sin embargo, creemos que esta tarea puede simplificarse mediante la creación de un controlador que se encargue de lanzar las etapas en diferentes hilos, para que sigan ejecutandose a la vez, y pararlas cuando el usuario lo requiera.

5.2. Estructura

La idea es generar una estructura que permita al usuario definir el orden de las etapas de manera sencilla, y una vez definidas las lance en diferentes hilos. Una posible representación del workflow sería un grafo acíclico dirigido, es importante que sea acíclico puesto que de otro modo la información nunca llegaría a un estado definitivo en el cual se acepte o rechace.

Podemos entonces definir cada fase como una entrada en un diccionario en la cual podemos almacenar la entrada y la salida, de este modo conectar una etapa con otra tan solo implica igualar la entrada de una a la salida de la otra. A la hora de lanzar el hilo es necesario saber a que función llamar para lo cual nos encontramos con dos opciones: llamar siempre a una función con un nombre determinado o simplemente incluir otro dato en el diccionario. En este caso se ha optado por que además de lo que ya guardaba se almacene la función a la que se debe llamar, esta opción da libertad a la hora de crear los pasos además de facilitar la posterior creación de los hilos.

Existe un último problema, algunas etapas del workflow pueden tener más de una entrada o salida por lo que en el diccionario se almacenara un array de entradas y otro de salidas, debido a esto también será necesario especificar el número de entrada o salida a la que se hace referencia al conectar los pasos.

Una entrada del diccionario tendría por tanto el siguiente formato.

```
flow[ 'step' ] = {
    'input': [
        flow[ 'other' ][ 'output' ][0] ,
        settings.directory
    ],
    'output': [
        settings.directory2
    ],
    'callback': step.run,
}
```

Además se hace uso de una etapa *dummy* que tan solo define la entrada del primer paso en su salida. Una vez creadas las entradas en el diccionario para todas las etapas solo será necesario ejecutar el workflow.

5.3. Etapas

Si bien es cierto que el workflow esta pensado para dar libertad a la hora de implementar las etapas tiene dos requisitos que se han de cumplir. El primero es a la hora de definir el método que se utilizara como callback en el diccionario, sus parámetros deben estar ordenados de modo que reciba primero todas las entradas, luego todas las salidas y por último un evento de la clase *threading*, este orden ha de ser tenido en cuenta a la hora de crear la entrada del diccionario descrita en la sección 5.2. En segundo lugar se debe asegurar que la etapa se mantendrá a la espera de nuevos datos que procesar hasta que se active el evento que se recibe por parámetro y que una vez recibido no se abortara la ejecución del paso hasta que sea seguro.

Para poder saber que está pasando el workflow también monta un sistema de log, utilizando la librería *logging* de python, al cual las etapas pueden

acceder para escribir mensajes de error o información de debug que le pueda ser de utilidad al usuario para conocer que ha ocurrido.

Debido a que las etapas se encuentran en continua ejecución podría darse el caso en el que los mismos datos se procesen varias veces, para prevenir esto las etapas que hemos implementado mantienen un log de historia que solo modifican ellas en el cual almacenan el identificador de los datos al acabar de procesarlos, una de las primeras comprobaciones al recoger datos de la entrada es comprobar si el id se encuentra en la historia del paso, de ser así los datos no se procesan.

5.4. Uso

Para utilizar el workflow son necesarios tres pasos.

En primer lugar modificar el fichero de configuración, *settings.py*, en el cual se encuentran almacenados todos los directorios de los pasos que hemos creado y los parametros de procesamiento que pueden modificarse de cada etapa.

Ir al workflow, *workflow.py*, y editar la función **define_flow()** para crear el diccionario de etapas tal y como se ha explicado en la sección 5.2.

Por último, ejecutar el workflow. Para parar el procesamiento basta con enviar una interrupción, Ctrl+C, en la consola en la cual se haya lanzado.

Capítulo 6

Procesamiento

RESUMEN: A lo largo de este capítulo se presentaran los diferentes procesos a los que se pueden someter los datos, entre ellos se encuentran transformaciones que ya no se emplean en nuestro detector pero que podrían ser útiles si se exploran nuevas formas de detección.

6.1. Recoloreado

Esta etapa emplea la biblioteca PIL, Python Imaging Library, para cargar las imagenes y crear el canvas sobre el que guardar el resultado.

A simple vista las imagenes obtenidas en la sección 4.2 están coloreadas y nos permiten apreciar las diferencias fácilmente. El problema surge cuando queremos que el programa haga las misma extrapolaciones que hace nuestro cerebro.

Con el fin de facilitar la comparación de los pixels la imagen pasa por un proceso de recoloreado que la transforma pixel a pixel.

Los colores que reconoce este proceso se definen mediante un diccionario que almacena el nombre y composición del mismo. Esto facilita la modificación de los grupos reconocidos de modo que un cambio en el coloreado inicial de las imagenes no supone un problema.

El problema de este proceso es definir cómo ha de interpretar la tonalidad del pixel, para atajarlo hemos empleado la distancia Manhatttan. Se trata de medir la distancia entre la composición del pixel y la del grupo, asumiendo que ambas tonalidades son RGB y que están almacenadas en triplas, de la forma (R, G, B), tan solo hemos de restar cada componente con su homóloga y sumar los resultados para obtener un valor. El grupo que resulte con menor valor será aquel al que pertenezca el pixel.

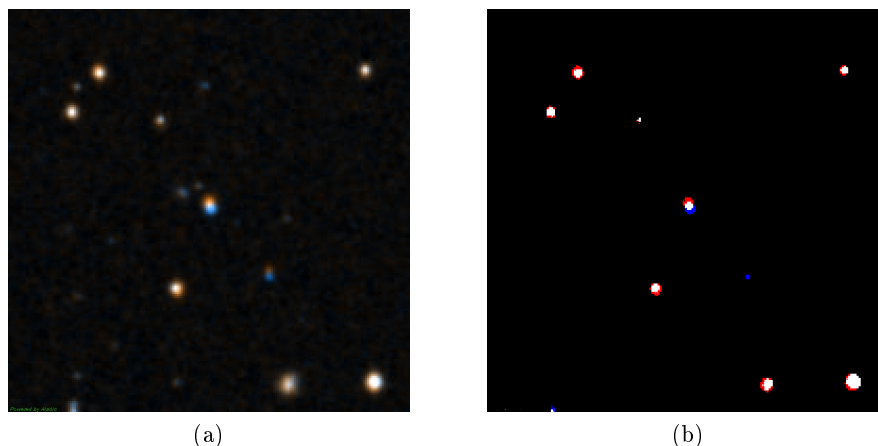


Figura 6.1: Recoloreado usando la distancia Manhattan

Tal y como muestra la imagen 6.1b hay información que se pierde durante el proceso debido a que las estrellas están muy apagadas haciendo que la distancia al negro sea menor que al azul o rojo. Para poder atajar este problema se realiza una comprobación posterior, si el grupo asignado es negro pero la distancia al azul o al rojo es inferior a un cierto umbral se reasigna el grupo del pixel, imagen 6.2.

Cuando se ha obtenido el grupo del pixel se pinta del color de grupo en un canvas, una vez se han procesado todos los pixel se almacena el resultado en un fichero png.

6.2. Comprobación

Aún cuando el detector acepta un sistema de estrellas dobles existe un problema, puede no tratarse de un sistema nuevo. Para solventar esto existe una fase de comprobación que se puede activar para todos los sistemas aceptados.

Si se ha solicitado esta fase se realiza un primer paso que consiste en conectarse a WDS y descargar los datos de los sistemas conocidos, este paso solo se hace una vez por ejecución, al comienzo, para no saturar los servidores. El hecho de emplear los mismos datos durante una ejecución completa no supone un problema puesto que la base de datos no se altera habitualmente.

Cuando un sistema es aceptado esta fase toma sus coordenadas y busca coincidencias en todas las entradas descargadas, si no hay ninguna se ignora el sistema detectado. Dejando por tanto un documento json de la siguiente forma.

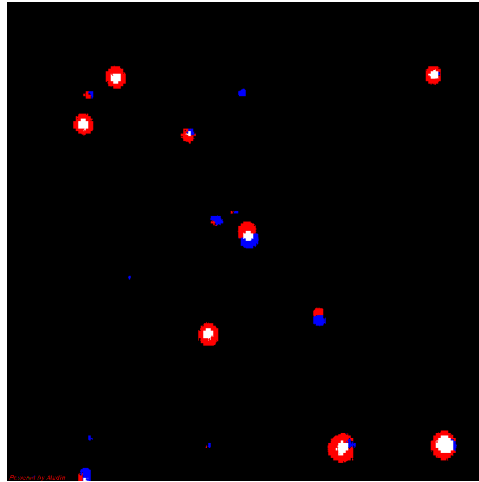


Figura 6.2: Recoloreado de la imagen 6.1a con un umbral de 150

```
{
  "1": {
    "Angle difference": 0.5332939070674456,
    "Separation difference": 6.23413572082471,
    "Maximum separation": 117.64777940955791,
    "Separation %": 5.298982906530099,
    "PA": 151.86140052006294,
    "Separation": 28.995116847884947,
    "Proper Motion A (brightest)": [
      163.2,
      -258.4
    ],
    "Proper Motion B": [
      176.79999999999998,
      -299.2
    ]
  }
}
```

Si por el contrario existen datos asociados al sistema se almacenan en el archivo json junto con los datos proporcionados por el detector y el calculo del error entre ambos, dando lugar a un archivo con el siguiente formato.

```
{
  "1": {
    "Angle difference": 1.419208447017608,
    "Separation difference": 3.3066248088095165,
    "Maximum separation": 123.22743201089601,
    "Separation %": 2.683351226955,
    "PA": 15.760824216099781,
```

```

    "Separation": 30.77825812822563,
    "Proper Motion A (brightest)": [
        136.0,
        0.0
    ],
    "Proper Motion B": [
        122.39999999999999,
        27.2
    ]
},
"wds": {
    "PA": 18.0,
    "Separation": 30.55,
    "Proper Motion A (brightest)": [
        93,
        15
    ],
    "Proper Motion B": [
        91,
        10
    ]
},
"error": {
    "PA": 2.239175783900219,
    "Separation": -0.22825812822562952,
    "Proper Motion A (brightest)": [
        -43.0,
        15.0
    ],
    "Proper Motion B": [
        -31.399999999999999,
        -17.2
    ]
}
}

```

Si bien esta fase es una primera comprobación sus resultados no son definitivos en el caso del no, puesto que solo se comprueban las coordenadas centrales de la imagen. Si la estrella no se encuentra en el centro es imposible para este programa determinar si el sistema ya se conocía o no.

6.3. Contador de pixels

Al igual que la etapa definida en la sección 6.1 se emplea la librería PIL. Esto no es lo único que comparten, el proceso al que se someten las imágenes es muy similar.

La idea de aplicar este proceso es obtener la composición de colores de

cada fotografía. Una vez obtenida se almacena el resultado en un fichero CSV que se puede analizar posteriormente. La gama de colores se define mediante un diccionario en el que se almacenan los colores puros que se quieren reconocer.

Una vez cargada la imagen se analiza cada pixel que la compone, para conocer el color, de entre los definidos en el diccionario, al que más se asemeja. como el nombre de la etapa indica el objetivo es contar los pixels, por lo tanto existe un contador por cada color en el cual se almacena el número de pixels de la imagen que pertenecen a dicha tonalidad.

Al terminar de procesar es posible guardar el valor de dichos contadores o transformarlos para obtener otros datos. En nuestro caso, decidimos almacenar los porcentajes de pixels rojos, azules y blancos, y las proporciones tanto de blancos como de azules con respecto de los rojos.

6.4. Corte

En esta etapa empleamos la librería `image_slicer` de Python, distribuida bajo licencia MIT. Esta librería permite dividir una fotografía en n piezas del mismo tamaño.

En este caso decidimos que las imágenes se dividieran en 9 sectores, lo cual permite reducir el nivel de ruido en la fotografía pero dejando datos suficientes en ella para posteriores análisis. En la figura 6.3 se puede observar que existe mucha información, sin embargo, si la dividimos en 9 fragmentos, figura 6.4, las imágenes resultantes tienen menos ruido.

Este proceso tan solo almacena el cuadrante central para reducir la carga de memoria, debido a esto solo la imagen de la figura 6.4e aparecería en la salida. Si esto quisiese modificarse tan solo habría que desactivar el parámetro `only_center` del método `run`.

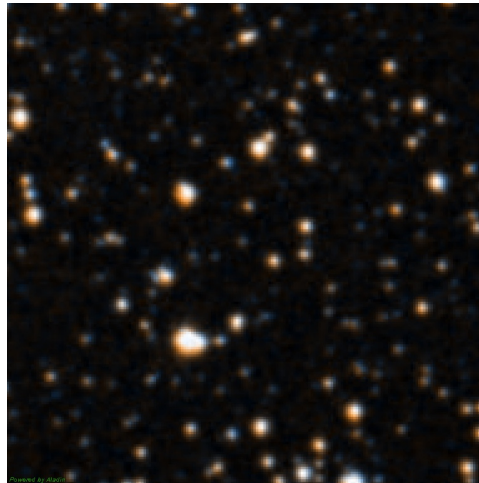


Figura 6.3: Imágen sin recortar

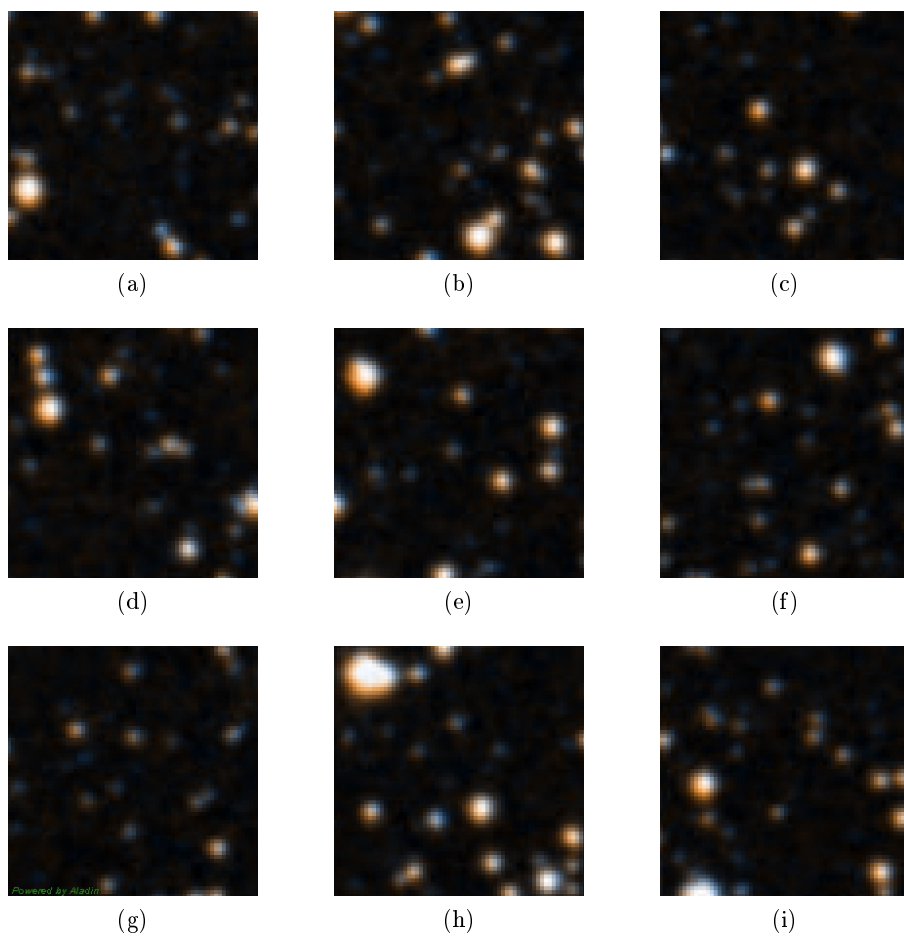


Figura 6.4: Imagenes resultantes de la división

Capítulo 7

Parametrización del detector

7.1. Introducción

...

Notas bibliográficas

...

Capítulo 8

Problemas

8.1. Introducción

...

Notas bibliográficas

...

Bibliografía

HARTKOPF, W., HARMANEC, P. y GUINAN, E. *Binary Stars as Critical Tools and Tests in Contemporary Astrophysics (IAU S240)*. Cambridge, 2007. ISBN 9780521863483.

LUYTEN, W. J. Double stars with common proper motion. 1971. Disponible en <http://adsbit.harvard.edu/full/1971Ap%26SS..11...49L> (último acceso, Mayo, 2018).

*–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

*–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

