shift

May 7, 2020

1 Shifting science frames

Written by Evgenii N.

The following code shifts the science frames so that stars appear at same x-y positions in all images. The input data is in `030_science_frames/data/reduced` directory and the shifted images are saved to `040_shift/data/shifted`.

1.1 Prerequisite code

[1]:

```
# Import libraries that we will use later in this notebook
import os
import shutil
import ccdproc
import numpy as np
from astropy.visualization import ZScaleInterval, MinMaxInterval, ImageNormalize
from astropy import units as u
from matplotlib.colors import LogNorm
from ccdproc import CCDData
import matplotlib.pyplot as plt
from photutils.aperture import CircularAperture, aperture_photometry
from photutils.centroids import centroid_2dg, centroid_com, centroid_1dg
from scipy.ndimage import shift

# Make images non-blurry on high pixel density screens
%config InlineBackend.figure_format = 'retina'

# Title size
plt.rcParams['axes.titlesize'] = 16

# Axes label size
plt.rcParams['axes.labelsize'] = 13


def show_image(image, title):
    """
    Display an image.
    Parameters
    ----------

    image: astropy.nddata.ccddata.CCDData
        A fits image to show.

    title: str
        Plot title.
    """
    fig, ax = plt.subplots(figsize=(12, 8))  # Change image size
    plt.rcParams.update({'font.size': 10})  # Change font size

    # Scale the image similar to 'zscale' mode in DS9.
    # This makes easier to spot things in the image.
    interval=ZScaleInterval()
    vmin, vmax = interval.get_limits(image)
    norm = ImageNormalize(vmin=vmin, vmax=vmax)

    plt.imshow(image, cmap='gray', norm=norm)  # Set color map and pixel scaling
    plt.xlabel('x [pixel]')  # Set axis labels
    plt.ylabel('y [pixel]')
    plt.title(title, y=-0.2)  # Set image title
    plt.colorbar()  # Show color bar


def print_image_stats(image, title):
    """
    Print first pixel value, average and standard deviation for an image.

    Parameters
    ----------

    image: astropy.nddata.ccddata.CCDData
        A fits image to show.

    title: str
        Image name.
    """

    data = np.asarray(image)# Get numpy array for image data
    label_len = 10  # Length of the text label
    first_pixel = data[0, 0]  # First pixel
    average = np.mean(data)  # Average
    standard_deviation = np.std(data)   # Standard deviation
```

```python
    # Print values
    # -------

    print(
        f'\n{title}',
        f"\n{'-' * len(title)}",
        f"\n{'Pixel:':<10}{first_pixel:>10.2f} ADU",
        f"\n{'Avg:':<10}{average:>10.2f} ADU",
        f"\n{'Std:':<10}{standard_deviation:>10.2f} ADU\n"
    )


def save_image(image, file_path):
    """
    Save image to disk. Overwrites the file if it already exist.

    Parameters
    ---------
    image: astropy.nddata.ccddata.CCDData
        Image to be saved

    file_path: str
        Path where the image is saved
    """

    # Delete the file if it already exists

    try:
        os.remove(file_path)
    except OSError:
        pass

    # Create directory
    # ------

    dirname = os.path.dirname(file_path)

    if not os.path.exists(dirname):
        os.makedirs(dirname)

    image.write(file_path)
```

1.2 Program code

[4]:

```
def load_images_from_dir(dir_path, include):
    """
    Loads fits images from the directory.

    Parameters
    ----------

    dir_path: str
        Path where the images are loaded from.

    include: str
        The pattern to filter the file names, e.g. '*.fit'.

    Returns
    --------
    (images, image_paths)
        images: list of astropy.nddata.ccddata.CCDData
            Loaded images.
        image_paths: list of str
            Image names.
    """

    images = ccdproc.ImageFileCollection(dir_path, glob_include=include)

    # Make sure we are only reading science images
    images = images.files_filtered(PICTTYPE=1)

    # Read the images
    return [
                CCDData.read(os.path.join(dir_path, image))
                for image in images
           ], images


def calculate_star_positions(images, box_center, box_size):
    """
    Calculate position of the star in `images` by searching
    within a box of size `box_size` centered at `star_position`.

    Parameters
    ----------

    images: list of astropy.nddata.ccddata.CCDData
        List of images.

    box_center: (x, y)
        Coordinates of the center of the search box.

    box_size: float
        The size of the search box.

    Returns
    --------
    list of (x, y)
        x, y: loat
            Positions of the star in the images.
    """

    half_box = int(box_size / 2)  # Half the size of the search box
    star_positions = []

    # Loop over the images
    for image in images:
        # Make copy of the image
        image = image.copy()

        # Subtract the background
        image = image - np.ma.median(image)

        # Get the image region where we will search for a star
        # Note: Y coordinates come first
        search_box = image[box_center[1] - half_box: box_center[1] + half_box,
                           box_center[0] - half_box: box_center[0] + half_box]

        # Estimate the pixel coordinates of the star, within the search box
        x_box, y_box = centroid_2dg(search_box)

        # Calculate x and y cooridinates of the star realtive to the image
        x = box_center[0] - half_box + x_box
        y = box_center[1] - half_box + y_box
        star_positions.append((x, y))

    return star_positions


def shift_image(image, xy_shift):
    """
    Shift one image.

    Parameters
    ----------
    image: astropy.nddata.ccddata.CCDData
        Images to be shifted.

    xy_shift: (dx, dy)
        Shift amount for the image.


    Returns
    -------
    astropy.nddata.ccddata.CCDData
        Shifted image.
    """

    # Get the star offset for this image relative to the first image
    # Note, x/y coordinates are flipped.
    yx_shift = (xy_shift[1], xy_shift[0])

    # Shift the image data
    # Use the median pixel value for the new pixels that appear after shifting
    # So it does not skew our background subtraction later
    shifted = shift(image, yx_shift, order=0, mode='constant', cval=np.
 ↪median(image))
    return CCDData(shifted, unit="adu")


def shift_images(images, shifts):
    """
    Shift the images.

    Parameters
    ----------

    images: list of astropy.nddata.ccddata.CCDData
        List of images to be shifted.

    shifts: list of (dx, dy)
        Shift amount for each image.


    Returns
    -------
    list of astropy.nddata.ccddata.CCDData
        Shifted images.
    """

    shifted_images = []
    for image, xy_shift in zip(images, shifts):
        shifted = shift_image(image, xy_shift)
        shifted_images.append(shifted)

    return shifted_images


def str_coordinate(position, decimal=0):
    """
    Formats a position to a string, e.g. "(12, 4)"

    Parameters
    ----------

    position: (x, y)
        x, y: float
            X-y coordinates

    decimal: float
        Number of decimal places to show.
    """
    return f'({position[0]:.{decimal}f}, {position[1]:.{decimal}f})'


def print_diagnistic(image_names, positions, shifts, shifted_positions,
                     max_shift):
    """
    Print the star's position and shifts before and after image.
    Display warning message if the star positions differ by more than max_shift␣
 ↪pixels
    between frames.

    Parameters
    ----------

    image_names: list of str
        Image file names.

    positions: list of (x, y)
        List of initial positions of the star before the shift.

    shifts: list of (dx, dy)
        List of shifts for the images.

    shifted_positions: list of (x, y)
        List of star's positions after the shift.
    max_shift: float
        Maximum position difference (pixels) for the star.
        If difference is larger, shows a warning message.
    """

    print(
        (
            f'{"Name":<40s}'
            f'{"Position":>10s}'
            f'{"Shift":>10s}'
            f'{"New shift":>10s}'
        )
    )

    print('-' * 75)
    shifted_positions = np.array(shifted_positions)
    new_shifs = shifted_positions[0] - shifted_positions

    for image_name, position, shift_amount, shifted_position \
        in zip(image_names, positions, shifts, new_shifs):

        warning = ''
        if abs(shifted_position[0]) > max_shift or abs(shifted_position[1]) >␣
 ↪max_shift:
            warning = '<----CHECK THIS'

        print(
            (
                f'{image_name:<40s}'
                f'{str_coordinate(position):>10s} '
                f'{str_coordinate(shift_amount):>10s} '
                f'{str_coordinate(shifted_position, decimal=1):>10s} '
                f'{warning}'
            )
        )


def pre_shift_images(images, image_names, pre_shift):
    """
    Shift images.

    Parameters
    ----------
    images: list of astropy.nddata.ccddata.CCDData
        Images to be shifted.

    image_names: list of str
```

```
        List of image file names.
    pre_shift: dict
        key: file name
        value: shift (dx, dy) to be done.

    Returns
    -------
    list of astropy.nddata.ccddata.CCDData
        Shifted images.
    """

    shifted_images = []

    for image, image_name in zip(images, image_names):
        if image_name in pre_shift:
            image = shift_image(image, pre_shift[image_name])

        shifted_images.append(image)


    return shifted_images


def remove_bad_files(dest_dir):
    """
    Remove images that were not shifted properly

    Parameters
    ----------

    dest_dir: str
        Output directory.
    """


    # Hardcoded list of bad images after looking at results in DS9
    bad_images = ['march_09_2018/NGC_3201_V_60.000secs_00001596.fit']

    for image_path in bad_images:
        full_path = os.path.join(dest_dir, image_path)

        if os.path.isfile(full_path):
            os.remove(full_path)


def shift_images_all_nights(all_shift_data, source_dir, dest_dir):
    """
    Shift images for all nights.

    Parameters
    ----------
    all_shift_data: list of dict
        Each dictionary has keys:
            'night': str
                The source directory.
            'box_center': (x, y)
                Position of the search box.
            'box_size': float
                Size of the search box.
            'pre_shift': dict
                Dict with file names as keys and (dx, dy) shifts as values.
    """

    for shift_data in all_shift_data:

        night = shift_data['night']
        print(f'\nShifting images in {os.path.join(source_dir, night):}\n')
        box_center = shift_data['box_center']
        box_size = shift_data['box_size']

        dir_path = os.path.join(source_dir, night)
        images, image_names = load_images_from_dir(dir_path=dir_path,␣
↪include='*.fit')

        # Make initial rough shift of the images
        # This is needed because some frames are shifted significantly
        images = pre_shift_images(images=images, image_names=image_names,
                                  pre_shift=shift_data['pre_shift'])

        # Detect a star within a box in each image and calculate its position
        positions = calculate_star_positions(images=images,
                                             box_center=box_center,
                                             box_size=box_size)

        # Calculate the shifts relative to the first image
        positions = np.array(positions)
        shifts = positions[0] - positions

        # Shift the images
        shifted_images = shift_images(images=images, shifts=shifts)

        # Calculate the star position in the shifted images
        shifted_positions = calculate_star_positions(images=shifted_images,
                                                     box_center=box_center,
                                                    box_size=box_size)

        # Show shifting numbers and check that the star's positions
        # after the shift differ by no more than 0.5 pixels
        print_diagnistic(image_names=image_names, positions=positions,␣
 ↪shifts=shifts,
                         shifted_positions=shifted_positions,
                         max_shift=0.5)

        # Save shifted images to disk
        # ---------

        for shifted_image, image_name in zip(shifted_images, image_names):
            dest_path = os.path.join(dest_dir, night, image_name)

            # Subtract background
            shifted_image -= np.ma.median(shifted_image)

            shifted_image = CCDData(shifted_image, unit="adu")
            save_image(image=shifted_image, file_path=dest_path)
```

[5]:
```
# Set Bias and Dark image paths
# --------

# Shifting settings for each night:
#     'night': str
#         The source directory for the night
#     'box_center': (x, y)
#         Position of the box where a star will be searched.
#     'box_size': float
#         Size of the search box.
#     'pre_shift': dict
#         Dictionary with file names as keys and (dx, dy) shifts as values.
#         Used for doing initial rough shift of the image, where needed.
all_shift_data = [
    {
        'night': 'march_29_2018',
        'box_center': (560, 426),
        'box_size': 30,
        'pre_shift': {}
    },
    {
        'night': 'april_30_2018',
        'box_center': (556, 410),
        'box_size': 25,
        'pre_shift': {}
    },
    {
        'night': 'march_09_2018',
        'box_center': (676, 333),
        'box_size': 25,
        'pre_shift': {
            'NGC_3201_B_60.000secs_00001613.fit': (-3, 5),
            'NGC_3201_B_60.000secs_00001652.fit': (-20, 15),
            'NGC_3201_I_60.000secs_00001581.fit': (0, -8),
            'NGC_3201_I_60.000secs_00001584.fit': (7, -3),
            'NGC_3201_I_60.000secs_00001593.fit': (1, -4),
            'NGC_3201_B_60.000secs_00001644.fit': (-17, 12),
            'NGC_3201_B_60.000secs_00001649.fit': (-22, 15),
            'NGC_3201_B_60.000secs_00001650.fit': (-22, 15),
            'NGC_3201_B_60.000secs_00001651.fit': (-22, 15),
            'NGC_3201_B_60.000secs_00001652.fit': (-22, 17),
            'NGC_3201_B_60.000secs_00001653.fit': (-22, 15),
            'NGC_3201_R_60.000secs_00001624.fit': (-10,  14),
            'NGC_3201_R_60.000secs_00001625.fit': (-10,  14),
            'NGC_3201_R_60.000secs_00001626.fit': (-10,  14),
            'NGC_3201_R_60.000secs_00001627.fit': (-10,  14),
            'NGC_3201_R_60.000secs_00001629.fit': (-10,  14),
            'NGC_3201_R_60.000secs_00001630.fit': (-10,  14),
            'NGC_3201_R_60.000secs_00001631.fit': (-10,  14),
            'NGC_3201_R_60.000secs_00001632.fit': (-10,  14),
            'NGC_3201_R_60.000secs_00001633.fit': (-10,  14),
            'NGC_3201_V_60.000secs_00001635.fit': (-13,  10),
            'NGC_3201_V_60.000secs_00001636.fit': (-12,  12),
            'NGC_3201_V_60.000secs_00001637.fit': (-13,  12),
            'NGC_3201_V_60.000secs_00001638.fit': (-13,  20),
            'NGC_3201_V_60.000secs_00001639.fit': (-13,  20),
            'NGC_3201_V_60.000secs_00001641.fit': (-13,  20),
            'NGC_3201_V_60.000secs_00001642.fit': (-13,  20),
            'NGC_3201_V_60.000secs_00001643.fit': (-13,  20)
        }
    }
]

source_dir = '../030_science_frames/data/reduced'
dest_dir = './data/shifted'

shift_images_all_nights(all_shift_data, source_dir=source_dir,␣
 ↪dest_dir=dest_dir)
remove_bad_files(dest_dir=dest_dir)

print("-------------------")
print("We are done")
```

```
Shifting images in ../030_science_frames/data/reduced/march_29_2018

Name                                       Position     Shift New shift
----------------------------------------------------------------------------
NGC_3201_B_30.000secs_00000472.fit       (561, 418)     (0, 0) (0.0, 0.0)
NGC_3201_B_5.000secs_00000458.fit        (559, 425)    (2, -7) (0.4, -0.5)
NGC_3201_B_5.000secs_00000459.fit        (559, 425)    (2, -7) (0.3, -0.1)
NGC_3201_B_5.000secs_00000460.fit        (559, 424)    (2, -6) (-0.2, -0.2)
NGC_3201_I_30.000secs_00000479.fit       (560, 412)     (1, 5) (-0.0, 0.3)
NGC_3201_I_30.000secs_00000480.fit       (563, 412)    (-1, 6) (-0.4, -0.2)
NGC_3201_I_5.000secs_00000467.fit        (557, 421)    (5, -4) (-0.4, 0.3)
NGC_3201_I_5.000secs_00000468.fit        (558, 421)    (3, -3) (0.3, -0.4)
NGC_3201_I_5.000secs_00000469.fit        (557, 421)    (5, -3) (-0.4, -0.5)
NGC_3201_R_30.000secs_00000476.fit       (557, 415)     (4, 3) (-0.1, 0.1)
NGC_3201_R_30.000secs_00000477.fit       (553, 414)     (9, 4) (-0.5, -0.4)
NGC_3201_R_5.000secs_00000464.fit        (553, 423)    (8, -5) (-0.2, -0.2)
NGC_3201_R_5.000secs_00000465.fit        (554, 423)    (7, -5) (0.3, -0.2)
NGC_3201_R_5.000secs_00000466.fit        (554, 423)    (7, -6) (-0.2, 0.5)
NGC_3201_V_30.000secs_00000473.fit       (561, 417)    (-0, 1) (-0.1, -0.2)
NGC_3201_V_30.000secs_00000475.fit       (556, 415)     (5, 3) (0.2, -0.2)
NGC_3201_V_5.000secs_00000461.fit        (557, 424)    (5, -7) (-0.3, 0.4)
NGC_3201_V_5.000secs_00000462.fit        (555, 424)    (7, -6) (-0.4, 0.0)
NGC_3201_V_5.000secs_00000463.fit        (554, 423)    (7, -5) (-0.1, -0.2)

Shifting images in ../030_science_frames/data/reduced/april_30_2018

Name                                       Position     Shift New shift
----------------------------------------------------------------------------
NGC_3201_B_30.000secs_00001305.fit       (552, 404)     (0, 0) (0.0, 0.0)
NGC_3201_B_30.000secs_00001306.fit       (555, 403)    (-3, 1) (0.2, -0.3)
NGC_3201_B_5.000secs_00001292.fit        (563, 410)  (-11, -7) (0.2, 0.4)
NGC_3201_B_5.000secs_00001293.fit        (563, 410)  (-11, -6) (-0.1, 0.0)
NGC_3201_B_5.000secs_00001294.fit        (563, 409)  (-11, -6) (-0.1, 0.4)
NGC_3201_I_30.000secs_00001313.fit       (552, 400)    (-0, 4) (-0.0, 0.1)
NGC_3201_I_30.000secs_00001315.fit       (561, 399)    (-9, 4) (0.2, 0.4)
NGC_3201_I_5.000secs_00001301.fit        (557, 408)   (-5, -4) (-0.0, -0.0)
NGC_3201_I_5.000secs_00001302.fit        (557, 408)   (-5, -4) (-0.3, 0.1)
NGC_3201_I_5.000secs_00001303.fit        (557, 406)   (-5, -2) (-0.1, -0.1)
NGC_3201_R_30.000secs_00001310.fit       (557, 402)    (-5, 1) (-0.3, 0.3)
NGC_3201_R_30.000secs_00001312.fit       (550, 401)     (2, 3) (-0.3, -0.1)
NGC_3201_R_5.000secs_00001298.fit        (558, 409)   (-7, -5) (0.4, 0.0)
NGC_3201_R_5.000secs_00001299.fit        (557, 409)   (-6, -5) (0.4, 0.3)
NGC_3201_R_5.000secs_00001300.fit        (558, 408)   (-6, -5) (0.1, 0.4)
NGC_3201_V_30.000secs_00001307.fit       (557, 403)    (-6, 1) (0.4, -0.4)
NGC_3201_V_30.000secs_00001308.fit       (563, 403)   (-11, 1) (0.1, -0.1)
NGC_3201_V_30.000secs_00001309.fit       (559, 403)    (-7, 1) (-0.5, 0.2)
NGC_3201_V_5.000secs_00001295.fit        (563, 409)  (-11, -5) (-0.3, -0.3)
```

```
NGC_3201_V_5.000secs_00001296.fit       (562, 409)  (-11, -5) (0.4, -0.1)
NGC_3201_V_5.000secs_00001297.fit       (561, 409)   (-9, -5) (0.0, -0.0)

Shifting images in ../030_science_frames/data/reduced/march_09_2018

Name                                      Position     Shift New shift
---------------------------------------------------------------------------
NGC_3201_B_60.000secs_00001604.fit      (674, 331)     (0, 0) (0.0, 0.0)
NGC_3201_B_60.000secs_00001605.fit      (675, 332)   (-1, -0) (0.0, -0.2)
NGC_3201_B_60.000secs_00001606.fit      (675, 331)    (-1, 1) (-0.0, -0.4)
NGC_3201_B_60.000secs_00001607.fit      (675, 331)    (-1, 1) (-0.3, -0.4)
NGC_3201_B_60.000secs_00001608.fit      (676, 331)    (-2, 1) (0.3, -0.3)
NGC_3201_B_60.000secs_00001609.fit      (676, 330)    (-2, 1) (-0.2, 0.1)
NGC_3201_B_60.000secs_00001610.fit      (677, 330)    (-2, 2) (-0.5, -0.3)
NGC_3201_B_60.000secs_00001611.fit      (677, 329)    (-3, 2) (0.0, -0.1)
NGC_3201_B_60.000secs_00001612.fit      (677, 329)    (-3, 2) (-0.2, 0.4)
NGC_3201_B_60.000secs_00001613.fit      (674, 334)   (-0, -2) (-0.3, -0.2)
NGC_3201_B_60.000secs_00001644.fit      (675, 332)   (-1, -0) (0.6, 0.3)
<----CHECK THIS
NGC_3201_B_60.000secs_00001649.fit      (677, 334)   (-3, -2) (-0.3, -0.4)
NGC_3201_B_60.000secs_00001650.fit      (676, 333)   (-2, -2) (0.3, 0.1)
NGC_3201_B_60.000secs_00001651.fit      (675, 333)   (-1, -1) (-0.2, -0.2)
NGC_3201_B_60.000secs_00001652.fit      (675, 334)   (-1, -3) (-0.1, 0.2)
NGC_3201_B_60.000secs_00001653.fit      (676, 331)    (-2, 0) (0.3, 0.0)
NGC_3201_I_60.000secs_00001581.fit      (673, 333)    (1, -2) (0.3, 0.5)
NGC_3201_I_60.000secs_00001584.fit      (675, 334)   (-1, -3) (-0.1, -0.0)
NGC_3201_I_60.000secs_00001585.fit      (669, 337)    (5, -6) (0.3, 0.2)
NGC_3201_I_60.000secs_00001586.fit      (669, 337)    (5, -5) (0.3, -0.2)
NGC_3201_I_60.000secs_00001587.fit      (670, 337)    (4, -5) (0.2, -0.4)
NGC_3201_I_60.000secs_00001588.fit      (670, 336)    (4, -5) (0.2, -0.1)
NGC_3201_I_60.000secs_00001589.fit      (671, 337)    (3, -5) (0.4, -0.2)
NGC_3201_I_60.000secs_00001590.fit      (671, 336)    (3, -5) (-0.4, 0.5)
NGC_3201_I_60.000secs_00001591.fit      (672, 336)    (3, -4) (-0.4, -0.5)
NGC_3201_I_60.000secs_00001592.fit      (672, 335)    (2, -4) (0.2, -0.1)
NGC_3201_I_60.000secs_00001593.fit      (673, 331)     (1, 0) (0.1, 0.3)
NGC_3201_I_60.000secs_00001615.fit      (680, 328)    (-6, 4) (0.5, -0.5)
NGC_3201_I_60.000secs_00001616.fit      (680, 328)    (-6, 3) (0.4, 0.4)
NGC_3201_I_60.000secs_00001617.fit      (680, 328)    (-6, 3) (0.1, 0.5)
NGC_3201_I_60.000secs_00001618.fit      (681, 328)    (-7, 4) (0.1, -0.5)
NGC_3201_I_60.000secs_00001619.fit      (682, 328)    (-8, 4) (0.1, -0.2)
NGC_3201_I_60.000secs_00001620.fit      (682, 327)    (-8, 4) (0.0, 0.3)
NGC_3201_I_60.000secs_00001621.fit      (683, 327)    (-8, 4) (-0.5, 0.4)
NGC_3201_I_60.000secs_00001622.fit      (683, 327)    (-9, 5) (0.1, -0.4)
NGC_3201_I_60.000secs_00001623.fit      (684, 327)   (-10, 5) (0.5, -0.4)
NGC_3201_R_60.000secs_00001563.fit      (670, 340)    (4, -9) (-0.3, -0.0)
NGC_3201_R_60.000secs_00001564.fit      (672, 341)    (2, -9) (-0.1, -0.4)
NGC_3201_R_60.000secs_00001565.fit      (674, 341)   (1, -10) (-0.5, 0.2)
NGC_3201_R_60.000secs_00001566.fit      (674, 341)   (0, -10) (0.4, 0.0)
```

```
NGC_3201_R_60.000secs_00001567.fit      (674, 341)    (0, -9) (0.2, -0.4)
NGC_3201_R_60.000secs_00001568.fit      (674, 341)   (0, -10) (0.4, 0.2)
NGC_3201_R_60.000secs_00001569.fit      (674, 341)  (-0, -10) (-0.1, 0.5)
NGC_3201_R_60.000secs_00001570.fit      (674, 340)   (-0, -9) (-0.3, 0.1)
NGC_3201_R_60.000secs_00001571.fit      (675, 340)   (-1, -9) (0.3, -0.1)
NGC_3201_R_60.000secs_00001624.fit      (674, 341)    (0, -9) (0.3, -0.2)
NGC_3201_R_60.000secs_00001625.fit      (674, 340)   (-0, -9) (-0.1, 0.2)
NGC_3201_R_60.000secs_00001626.fit      (674, 339)   (-0, -8) (-0.2, -0.0)
NGC_3201_R_60.000secs_00001627.fit      (675, 340)   (-1, -8) (0.2, -0.3)
NGC_3201_R_60.000secs_00001629.fit      (675, 338)   (-1, -7) (-0.3, 0.1)
NGC_3201_R_60.000secs_00001630.fit      (675, 338)   (-1, -6) (0.0, -0.5)
NGC_3201_R_60.000secs_00001631.fit      (675, 337)   (-1, -6) (0.2, -0.1)
NGC_3201_R_60.000secs_00001632.fit      (676, 337)   (-1, -6) (-0.4, 0.3)
NGC_3201_R_60.000secs_00001633.fit      (676, 337)   (-2, -6) (0.3, 0.4)
NGC_3201_V_60.000secs_00001594.fit      (671, 334)    (3, -3) (-0.1, -0.0)
NGC_3201_V_60.000secs_00001595.fit      (671, 334)    (3, -3) (-0.3, 0.3)
NGC_3201_V_60.000secs_00001596.fit      (672, 334)    (2, -3) (-0.0, 0.4)
NGC_3201_V_60.000secs_00001597.fit      (671, 334)    (3, -2) (-0.3, -0.2)
NGC_3201_V_60.000secs_00001599.fit      (674, 334)    (0, -2) (0.5, -0.4)
NGC_3201_V_60.000secs_00001600.fit      (674, 333)    (0, -2) (0.2, -0.0)
NGC_3201_V_60.000secs_00001601.fit      (674, 333)    (0, -2) (0.4, 0.5)
NGC_3201_V_60.000secs_00001602.fit      (674, 332)   (-0, -1) (-0.0, 0.0)
NGC_3201_V_60.000secs_00001603.fit      (674, 332)   (-0, -1) (-0.1, 0.3)
NGC_3201_V_60.000secs_00001635.fit      (674, 332)    (1, -0) (-0.5, -0.5)
NGC_3201_V_60.000secs_00001636.fit      (675, 333)   (-1, -2) (0.3, 0.2)
NGC_3201_V_60.000secs_00001637.fit      (674, 333)    (0, -2) (0.0, 0.3)
NGC_3201_V_60.000secs_00001638.fit      (676, 341)  (-2, -10) (0.5, 0.2)
NGC_3201_V_60.000secs_00001639.fit      (675, 340)   (-1, -9) (-0.1, 0.0)
NGC_3201_V_60.000secs_00001641.fit      (677, 340)   (-3, -9) (-0.2, 0.3)
NGC_3201_V_60.000secs_00001642.fit      (678, 340)   (-4, -8) (0.1, -0.4)
NGC_3201_V_60.000secs_00001643.fit      (679, 340)   (-5, -8) (0.3, -0.3)
------------------
We are done
```

1.3 Checking the results

I manually check all shiften frames in DS9 to see if they were shifted correctly. This video shows frames from March 9:

https://youtu.be/Z9XV4Pqw8lE

The stars appear at approximatelly same positions, I'm happy with the result.

1.4 Subtracting background

Note that I also subtracted the background (image median value) from the shifted images before saving.

1.5 Problems I found

When writing the shifting code I found that `centroid_2dg(search_box)` function could not find star position in a given box. The reason was that some raw frames were shifted too much, maybe because of the problems with telescope tracking. Solution was to do rough pre-shifting, see the `pre_shift` in the code.

1.6 Shifting settings

I just wanted to point out that one needs to be careful when using function shift, which shifts the image:

shifted = shift(image, yx_shift, order=0, mode='constant', cval=np.median(image))

With order=0 parameter (the one we used in Lab 3) it shifts the image by integer number of pixels, even if we specify fractional shifts, like 0.5. After I changed that to order=1, I noticed that now the image can be shifted by fractional number of pixels. One would think that this is better, however, it requires using interpolation to calculate the pixel values. And this can potentially affect the calculated fluxes. So we need to be careful. That's why I ended up keeping order=0 setting we used in the lab. I've made animated gifs comparing order=0 and order=1 settings here:

https://github.com/evgenyneu/asp3231_project/tree/master/code/040_shift

We can see that order=1 does change the pixel values, and I would avoid that.

[ ]: