

Multiview Implementation in Open CV-based Crescent Observation Application

Adi Damanhuri^{1*}, Sam'un², Agus Solikin³

¹UIN Sunan Ampel Surabaya

(Jl. Ahmad Yani No.117, Jemur Wonosari, Kec. Wonocolo, Surabaya, Jawa Timur 60237)

*Email: adidamanhuri@uinsby.ac.id

Abstract

Determining the beginning of the Hijri month is very important for Muslims, especially in Indonesia. One way to determine the beginning of the month is by observing the new moon or moon sighting. In the process of hilal (crescent) sighting, there are theoretically obstacles related to the validity of the sighting results. By implementing an observation system that has the function of displaying and storing observations through a developed application, it is hoped that it can help to increase the validity of the new moon observation. This study answers two problem formulations, specifically: (1) Can an OpenCV-based new moon observation application be built? (2) Can the multiview function be implemented with OpenCV? By using the Python programming language, the OpenCV library, and other additional libraries, an OpenCV-based new moon observation application has been successfully created, and at the same time the Multiview function has been implemented. The used application works well, especially the multiview function and the function to save the observations in the specified image format.

Keywords: Crescent Observation, Python, Opencv, Multiview

Article Info

Received:

08 November 2023

Revised:

17 November 2023

Accepted:

28 Desember 2023

Published:

30 Desember 2023

A. Introduction

The determination of the beginning of the lunar month is a duty of Muslims [1]. Determining the Hijri months means to determine the time of worship within the Hijri months including the sets of the beginning of Ramadan, the beginning of Shawal--which is also the time to end fasting in Ramadan, the determination of Zulhijja to set the day of fasting in Zulhijja 9th as well as the time of Wuquf in Arafat, the day of Eid al Adha and also the Tashreeq days. In fact, it is no less important, for example, to determine the beginning of the month of Muharram to mark the new Hijri year, and also for other months that correspond to determining the dates for performing Sunnah worship, for example, the fast of Yaumul Bidh, which is performed on the 14th and 15th of each month.

Of the 12 Hijri months, the most noteworthy for the Muslim community in general is the determination of the 3 crucial months, which are the beginning of Ramadan, the beginning of Shawwal and the beginning of Zulhijja. Indonesia, in particular, is not only concerned with the aspect of worship in determining these three crucial months, but there are other aspects and implications such as social, economic and even political. The emergence of multiple aspects in determining the 3 crucial lunar months is due to the fact that in these months there are mass worships and celebrations, such as the obligatory fast in the month of Ramadan, which is observed for a whole month, Eid al-Fitr at the beginning of Shawwal, which marks the end of the Ramadan fast, Hajj as well as Eid al-Adha in the month of Zulhijja. There is a culture in particular in determining the beginning of the month of Shawwal or Eid al-Fitr that is very inherent to the people of Indonesia, that is, the migration of people from the cities to the villages, which is actually a temporary visit back to their hometowns to gather and meet with extended family when celebrating Eid al-Fitr, known as *Mudik Lebaran* (Homecoming). Since it is a massive movement, it obviously needs a complex arrangement in terms of transport, financial availability and other related things.

The unique and special things about determining the beginning of the Hijri months are the differences in determining the beginning of the months and also the polarisation between the *Hisab* and *Rukyat* methods.

Especially for the determination of the beginning of the month by the Rukyat method, or the method whose practitioner ultimately relies on whether the crescent is visible or not, some very crucial cautions arise among activists and academics of astronomy, including the validity of the results of Rukyat. Nowadays, with the development of technology and

astronomical observation instruments, crucial notes regarding the validity of rukyat results can be answered by implementing an observation system that produces evidence of observation in the form of images of crescent observation results that can be viewed and stored as evidence of valid crescent or rukyat observation results. Observation systems that have used telescopes, telescope mounts, and detector cameras have been widely developed [2], but on general, it uses the camera control applications, with built-in applications from camera manufacturers, as well as paid applications that have certain functions that support the success of the observed hilal. These functions include the ability to access the detector camera used, the ability to view the image, the ability to modify the image and the ability to save the observed image [3].

The camera control application that has the supporting features for observing the crescent are mostly paid applications. In addition to the cost of the application, many of these applications do not have the ability to display observation images with multiple image profile views that can be displayed simultaneously.

With the availability of the Open Source Computer Vision Library or known as OpenCV, this research attempts to build an application that has functions that support the success of crescent observation and is able to display multiple image profiles simultaneously without the need to use multiple detector cameras. The Multiview function is important to be used to directly view 2 or more image profiles between the original or unmodified image and the modified image as evidence and reinforcement of the validity of the crescent observation or *Rukyat* results..

Based on the background described above, this research attempts to answer several problem formulations, including: (1) Can OpenCV-based hilal observation applications be built? (2) Can multiview features be implemented with OpenCV?

In this research, the author identifies several limitations, including the ability to build crescent observation applications with OpenCV and the ability of OpenCV to implement the multiview feature in the developed application. The purpose of this research is to build a crescent observation application that has a multiview function or can display multiple image profiles simultaneously to support the validity of the crescent observation results. The benefits of this research include adding options for the general public or crescent observation practitioners who want to implement a crescent observation system with high validity of observation results. In addition, by developing an open-source based crescent observation application, it is ensured that the developed application has no copyright problems, as the

developed application is indeed available to the general public openly and free of charge.

B. Research Methods

This research adopts the research and development or RnD research method. The data was taken from journal articles, books, popular writings and freelance work. Other data came in the form of application designs, software supporting application development, applications developed, and observational images.

C. Results and Discussion

Results

1. A Necessary Application

To build an OpenCV based crescent observation application with a multiview function, several applications are needed, including Python programming language, PyCharm IDE and several libraries.

a) Python Programming Language

Python is a very popular high-level programming language, famous for its easy to understand and easy to use syntax. It was developed by Guido van Rossum and first released in 1991. Python supports a variety of programming paradigms, including object-oriented programming, functional programming and script-based programming [4].

Python has versions 2.x and 3.x. Currently, the latest version of Python is Python 3, and most communities and projects use Python 3 as support for Python 2 ends on 1 January 2020 [4]. Python has become one of the most popular programming languages due to its wide range of uses, ease of learning, and the many communities and learning resources available online.

b) IDE Pycharm

PyCharm is not a compiler, but rather a very popular integrated development environment (IDE) for developing applications using the Python programming language. PyCharm is developed by JetBrains and offers various features to help developers write, edit, manage and test Python code more efficiently [5].

c) OpenCV

OpenCV (Open Source Computer Vision Library) is a real-time and dynamic image processing software library created by Intel and now supported by Willow Garage and Itseez.

It is freely available under the open source umbrella of the BSD licence. This library can be used for all platforms or as a cross-platform library. This programme is provided and mainly dedicated to real-time image processing [6].

d) Supporting Library

Several supporting libraries are needed to develop an OpenCV-based hilal observation application with Multiview, including

- (1) Matplotlib, is a Python library with a focus on data visualisation such as plotting graphs. Matplotlib was first coined or created by John D. Hunter. Matplotlib was originally designed to produce graphical plots suitable for journal publications or scientific articles. Matplotlib can be used in Python scripts, Python and IPython shells, web application servers, and several other graphical user interface (GUI) toolkits [7].
- (2) Numpy is one of the basic libraries or packages for scientific computing with Python. It is a Python library that provides multidimensional array objects, various derived objects (such as arrays and matrices), and a set of routines for fast operations on arrays, including mathematics, logic, shape manipulation, sorting, selection, I/O, discrete Fourier transform, basic linear algebra, basic statistical operations, random simulation, and more [8].
- (3) Time is a special module for time. Python programs can handle dates and times in different ways. Conversion between date formats is a common task for computers. Python's time and calendar module keeps track of dates and times [9].

2. Required Functions

a) Camera Access

The camera access function is the first and main feature before other features. When the camera access function is executed, other functions are executed. The camera access function is performed using OpenCV with the Python programming language to capture video from the camera and perform image processing or other analysis.

In the camera access script, the author use the `cap=cv2.VideoCapture(0)` function to create an object such as a VideoCapture object that will access the camera with index 0 (usually the default camera). In practice, some camera manufacturers do not allow access to cameras in applications built with OpenCV. Cameras with UDC format can be accessed with this OpenCV based application.

b) Signal Access

The script or code `cap.read()` is used to read the signal from the camera connected to the programme code.

c) Image Conversion

Image conversion in this research means changing the colour format of the image read by the camera. OpenCV provides several functions to perform image conversion, including colour format change, histogram processing and others.

The code or script to convert an image from colour format to greyscale format by creating a new object called `gray` is as follows: `gray = cv2.cvtColor(cap, cv2.COLOR_BGR2GRAY)`.

d) Image Modification

The image modification function is applied to increase the contrast of the resulting crescent image, making it easier to detect. The image modifications applied include histogram equalisation, canny edge detection and frame stacking.

Histogram equalisation is applied to the greyscale image using the code or script `eh=cv2.equalizeHist(cap)`. While Canny Edge Detection is applied to perform edge detection with the code `can=cv2.Canny(cap, 100, 200)`, where 100 is the lower limit of the threshold and 200 is the upper limit of the threshold.

e) Displaying the Image

In the function of displaying the image, this is the uniqueness and main purpose of this research, with the application built, it can display images simultaneously for several image profiles at once. Generally, the code to display the image is `cv2.imshow(object source)`, because `gray` is the source of the image object for the greyish scale, which in this study is considered as a display for raw data, `eh` is the source of the image object with an image that has been modified in the form of histogram equalisation, and `can` is the image object with the canny edge detection function. So to display it, the code is written sequentially: `cv2.imshow(grey)`, `cv2.imshow(eh)` and `cv2.imshow(can)`.

f) Saving the Image

Another very important function to support the validity of the crescent observation results is the function to save the observed image. In OpenCV this function is usually written as `cv2.imwrite('name.PNG', source)`.

Here `'name.PNG'` is the name of the file to save and its format. For example, `'hilar.PNG'` means that the observed file with the name `'hilar'` will be saved with the PNG

image format, while *'source'* indicates the source object to be saved. To write an image with a grey image object named *'gray'* and in PNG format, the code would be *cv2.imwrite('gray.PNG', gray)*.

Discussion

Basically, applications are developed using the following scripts:

```
import cv2
cap = cv2.VideoCapture(0)
while (True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    eh = cv2.equalizeHist (gray)
    can = cv2.Canny (gray, 100, 200)
    cv2.imshow ('Abu-Abu', gray)
    cv2.imshow('HE',eh)
    cv2.imshow('Canny', can)
    cv2.imwrite('grey.PNG', gray)
    cv2.imwrite('eh.PNG',eh)
    cv2.imwrite('can.PNG', can)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

The script is run using the PyCharm Community IDE to see if the script can run properly or if it encounters any constraints. In the PyCharm IDE, constraints can be detected directly, as shown in Figure 1.

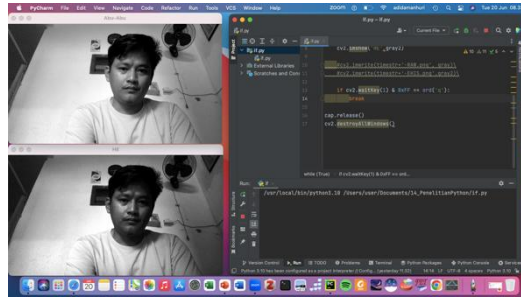


Figure 1 Running scripts in the PyCharm IDE

In the crescent observation on 19 July 2023, the multiview function can be performed well by the OpenCV-based crescent application, as shown in Figure 2..

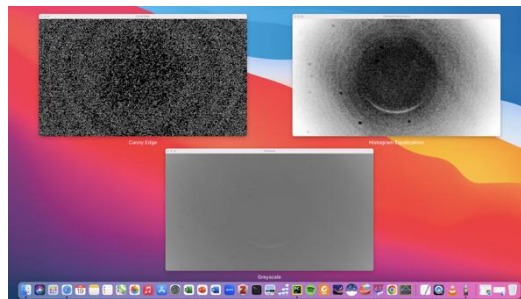


Figure 2 Multiview function display

The image results obtained by the OpenCV-based crescent observation application with the multiview function for the Hilal observation on 19 July 2023 are shown in Figure 3 for the greyscale image profile and in Figure 4 with the image profile modified by histogram equalisation.



Figure 3 Observed image with grey scale profile

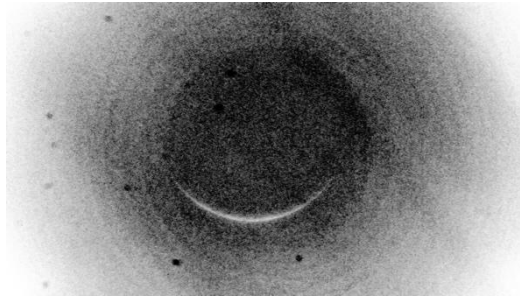


Figure 4 Observed image with histogram equalisation profile

D. Conclusion

The construction of an OpenCV based hilal observation application with the multiview function can be carried out and worked well. The multiview function together with the image saving function can help to increase the validity and inclusion of observational evidence of the crescent observation process. The barrier to research is the difficulty in finding a camera with an ideal resolution that can be accessed by an application built with OpenCV.

For crescent observers, OpenCV-based crescent observation applications with multiview capabilities can be used to optimise the observation process. For astronomical camera manufacturers, it can be considered to provide camera products with the ideal resolution for observation that can be accessed by OpenCV applications.

Acknowledgement

Our gratitude goes to UIN Sunan Ampel Surabaya through UINSA Institute for Research and Community Service for the Research Funding Assistance for the Basic Research Cluster of Study Programme Development for the fiscal year 2023 based on Rector's Decree Number 180 of 2023. We hope that the research center can continue to fund more research that we will compile in the future year.

References

- [1] Damanhuri, Adi, 'DESAIN SISTEM PENGAMATAN SABIT BULAN DI SIANG HARI', in *Semnastek*, 2015
- [2] Hidayat, T., P. Mahasena, B. Dermawan, D. Herdiwijaya, H. Setyanto, M. Irfan, and others, 'Developing Information System on Lunar Crescent Observations', *ITB Journal of Science*, 42 A.1 (2010), 67–80 <https://doi.org/10.5614/itbj.sci.2010.42.1.6>

- [3] Damanhuri, Adi, 'Sistem Pengamatan Hilal ISRN UHAMKA', *Al-Marshad: Jurnal Astronomi Islam Dan Ilmu-Ilmu Berkaitan*, 5729.June (2018), 1–11
<https://doi.org/10.30596/jam.v4i1.1931>
- [4] 'BeginnersGuide - Python Wiki' <<https://wiki.python.org/moin/BeginnersGuide>> [accessed 11 August 2023]
- [5] 'PyCharm: The Python IDE for Professional Developers by JetBrains', *JetBrains* <<https://www.jetbrains.com/pycharm/>> [accessed 11 August 2023]
- [6] IdMetafora.com, 'Mengenal OpenCV Dalam Python: Pengertian , Sejarah, Dukungan Pada OS, Fitur-Fitur, Jasa Pembuatan Website - Metafora Indonesia Tehnology' (idmetafora.com, 2014) <<https://idmetafora.com/news/read/1177/Mengenal-OpenCV-Dalam-Python-Pengertian-Sejarah-Dukungan-pada-OS-Fitur-fitur.html>> [accessed 11 August 2023]
- [7] 'Matplotlib — Visualization with Python' <<https://matplotlib.org/>> [accessed 11 August 2023]
- [8] 'NumPy' <<https://numpy.org/>> [accessed 11 August 2023]
- [9] Belajarpython, 'Belajarpython – Situs Open Source Tutorial Pemrograman Python Bahasa Indonesia.' <<https://belajarpython.com/tutorial/tanggal-waktu-python/>> [accessed 11 August 2023]