

How to install and use HOLO (.holo) file readers and writers for Python, ImageJ and Matlab

April 24, 2021

[About the scripts](#)

[Installation from the GitHub repository](#)

[holo file specification](#)

[Python scripts](#)

[holo file reader](#)

[holo file writer](#)

[Convert .holo files to movies or raw data](#)

[Matlab scripts](#)

[holo file reader](#)

[holo file writer](#)

[ImageJ plugins](#)

[holo file reader](#)

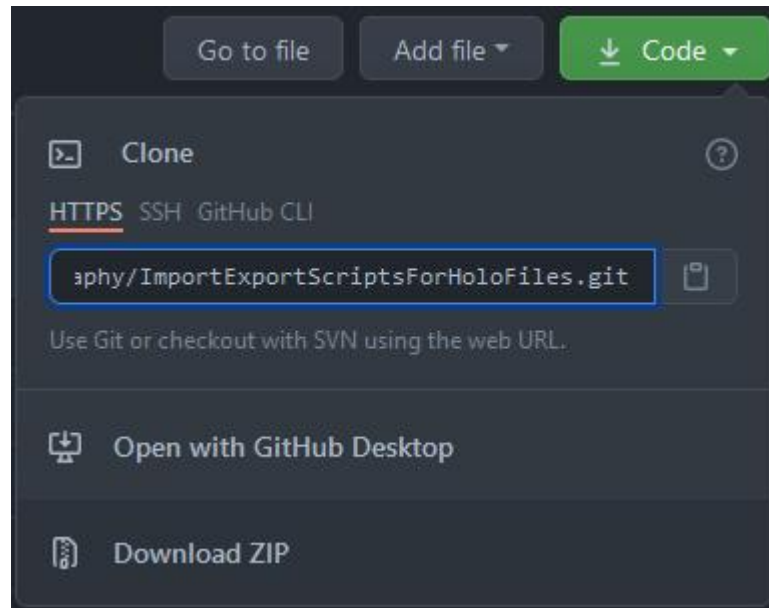
[holo file writer](#)

[Compiling .java files to .class to customize the plugins](#)

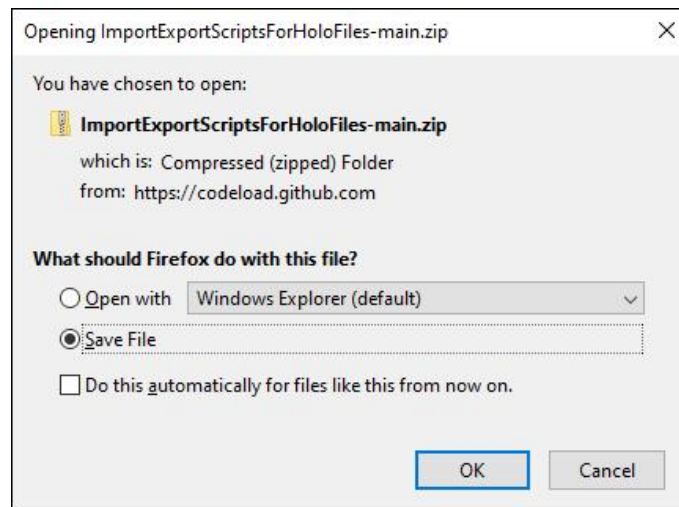
About the scripts

Installation from the GitHub repository

Download the collection of scripts and programs from github.com/DigitalHolography by cloning or clicking on “Download ZIP” at the “ImportExportScriptsForHoloFiles” folder from the [github repository](#).



Save the zip file that contains the collection of scripts and programs to read and write .holo files under ImageJ, Python, and Matlab.



holo file specification

If you want to learn more about the .holo structure, please read the [HOLO file specification](#).

Python scripts

The python folder contains a few scripts but we will focus here on the holo.py script. Import holo.py in the python file

containing your code using “import holo”. You can read and write HOLO files.

holo file reader

`holo.FileReader('filepath')`

`header, data, footer = holo.FileReader('filepath').get_all()`

Read an existing .holo file, ‘filepath’, and return all file data (‘header’, ‘data’, ‘footer’).

`data = holo.FileReader('filepath').get_all_frames()`

Read an existing .holo file, ‘filepath’, and return the ‘data’ part in bytes from the file.

holo file writer

`holo.FileWriter('filepath')`

`holo.FileWriter('filepath', (width, height, bytes_px, num_frames), data).write()`

Write a .holo file, ‘filepath’ is where the data will be written.

Width, height, bytes_px and num_frames are the characteristics of the .holo file you want to write and data is the value array (2D or 3D).

```
import holo

holo.FileWriter('test.holo', (2048,2048,1,4), data).write()
```

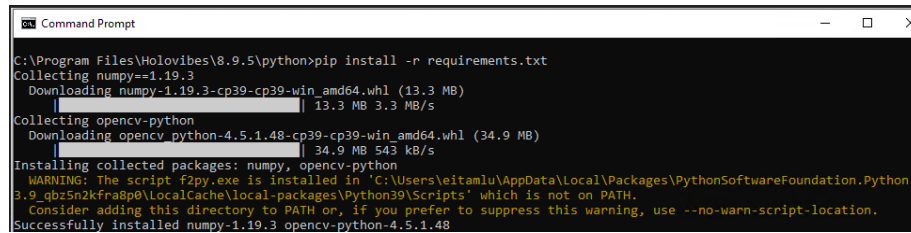
Convert .holo files to movies or raw data

In the “python” folder inside the Holovibes installation folder (C:\Program Files\Holovibes\9.0\python\), the

“convert_holo.py” script can be used to convert .holo files from and to different file formats. For that, have [python 3](#) installed, and install dependencies with “pip install -r requirements.txt” (the file requirements.txt can be found in the “python” folder of Holovibes). In the windows search tab, type “Windows PowerShell” or “Command prompt” and run it.



In the PowerShell or the Command Prompt, type : “pip install -r requirements.txt”.



```
Command Prompt
C:\Program Files\Holovibes\8.9.5\python>pip install -r requirements.txt
Collecting numpy==1.19.3
  Downloading numpy-1.19.3-cp39-cp39-win_amd64.whl (13.3 MB)
    13.3 MB 3.3 MB/s
Collecting opencv-python
  Downloading opencv-python-4.5.1.48-cp39-cp39-win_amd64.whl (34.9 MB)
    34.9 MB 543 kB/s
Installing collected packages: numpy, opencv-python
WARNING: The script f2py.exe is installed in 'C:\Users\gitamlu\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\localcache\local-packages\Python39\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.19.3 opencv-python-4.5.1.48
```

You must provide file extensions to get the expected conversion. The supported file types and extensions are holo, raw, avi, mp4, mkv :

- holo2avi

python3 convert_holo.py input.holo output.avi

- holo2mp4

python3 convert_holo.py input.holo output.mp4

- holo2mkv

python3 convert_holo.py input.holo output.mkv

- holo2raw

python3 convert_holo.py input.holo output.raw

- raw2holo

python3 convert_holo.py input.raw output.holo

A default video runs at 20fps. However, the video fps can be specified with the option --fps.

Example: **python3 convert_holo.py “input.holo” “output.mp4” --fps 60** (the output video fps will be 60 fps).

Sometimes people don’t have writing access to the directory “C:\Program Files\Holovibes\9.0\python\”, in that case, an error message may be obtained while trying to use the script “convert_holo.py” : “PermissionError: [Errno 13] Permission denied: 'test.raw'”. To overcome the issue, either get admin rights or simply try to move and use the scripts somewhere else on your computer.

Matlab scripts

In the Matlab folder you will find the scripts Read_Holo.m and Write_Holo.m.

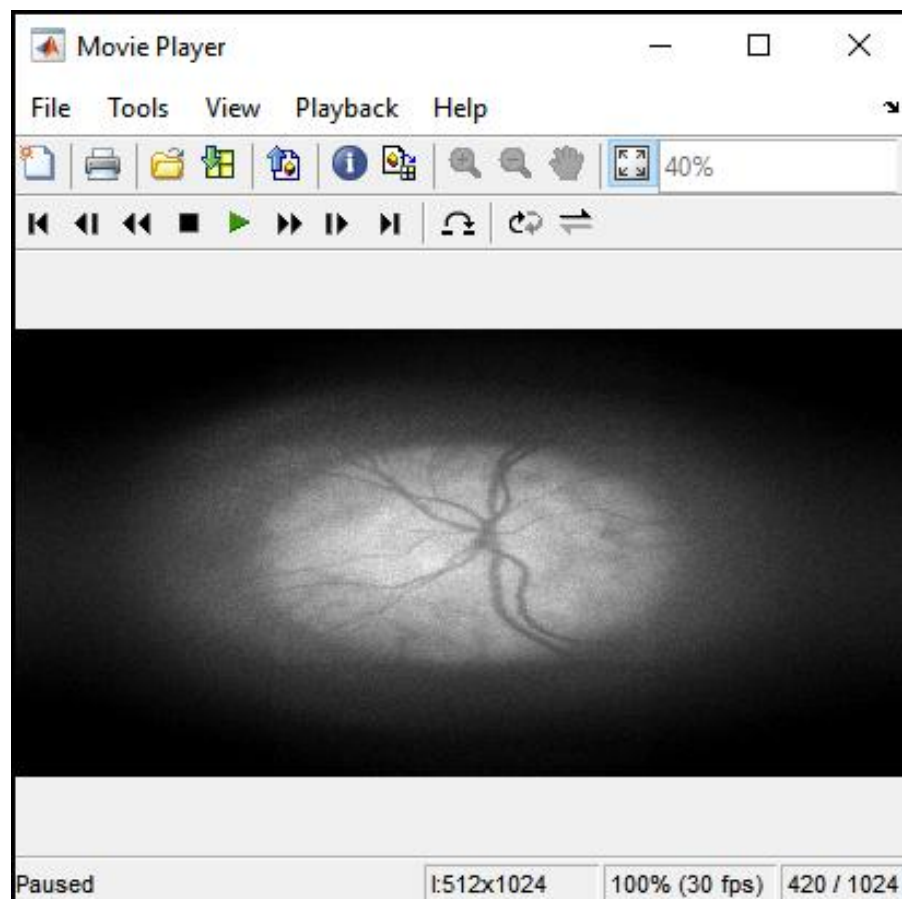
holo file reader

The matlab script Read_Holo.m file is a function which can be used in different ways. To test this function, you can download [this test file](#).

- Read_Holo()

Open the Read_Holo.m file and click on the “Run” button or write “Read_Holo” in the Command Window of matlab.

Choose the .holo file you want to watch with the dialog window. When the Movie Player window opens you can watch the sequence of images.



- Read_Holo('filepath')

Give the file path to the Read_Holo function and watch the video. Write on the “Command Window” or in your program

the command line : “Read_Holo(*filepath*)”. The “Movie Player” window will open.

- `output = Read_Holo()`

The dialog box will open. Choose the .holo you want to read.

The data from the .holo file will be saved in a “Output” variable but the “Movie Player” window will not open.

- `output = Read_Holo(filepath)`

Choose the .holo you want to read by giving directly the

“*filepath*” to the Read_Holo function. The data from the .holo file will be saved in a “Output” variable but the “Movie Player” window will not open.

holo file writer

The matlab script Write_Holo.m file is a function which can be used in different ways. It can be used in two different ways.

The purpose of this function is the writing of an array (2D or 3D) in a .holo file. To test this function, you can download this test file.

- `Write_Holo(Input, filepath)`

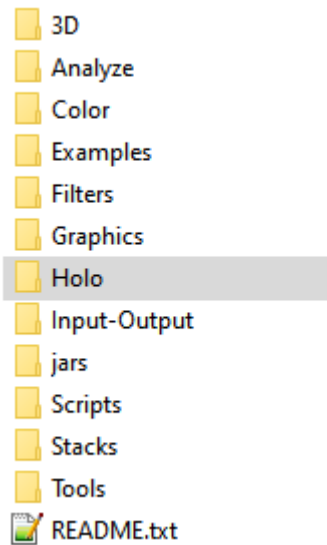
Give an “Input” array to the Write_Holo function and the “*filepath*” where the .holo file will be written.

- `Write_Holo(Input)`

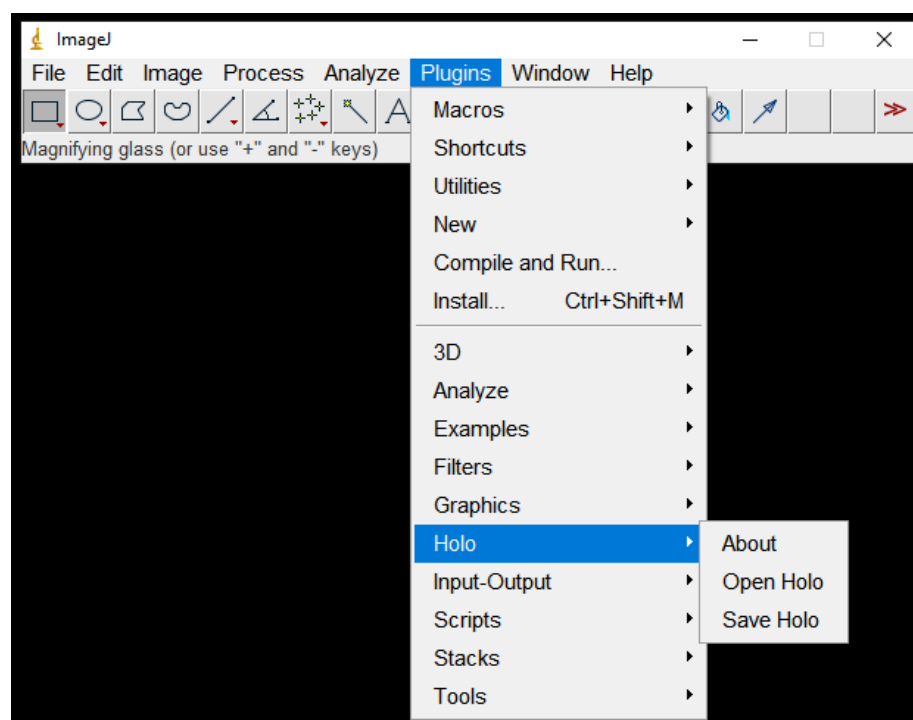
Give only the “Input” array after that the dialog window will open. Choose the location where the .holo file will be saved.

ImageJ plugins

First, download the [ImageJ software](#) (not Fiji, for which Holostuff plugins will not work out-of-the-box) and install it on your computer. Download the “Holo” folder from the [holostuff github repository](#). This folder contains .class and .java files. The java files are the source code for the .class files. Move this folder to the “ImageJ/plugins/” folder.



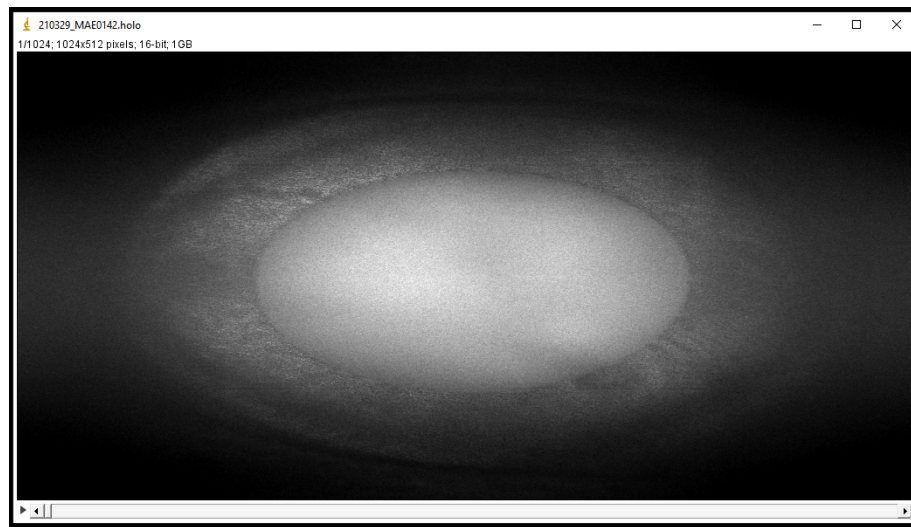
Restart ImageJ if it was open. Click on the “Plugins” menu where you will find new plugins for ImageJ gathered in the folder “Holo”.



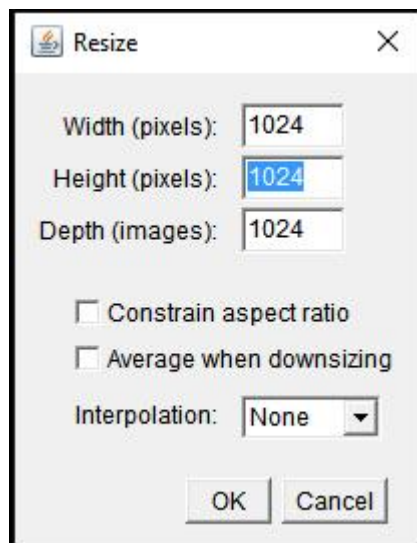
holo file reader

You can test the “Open Holo” plugin by downloading the file [210329_MAE0142_rendered_images.holo](https://doi.org/10.210329/MAE0142_rendered_images.holo). Click on the “Open Holo” plugin in ImageJ and select the HOLO file that you have just downloaded. The following window displaying an eye will

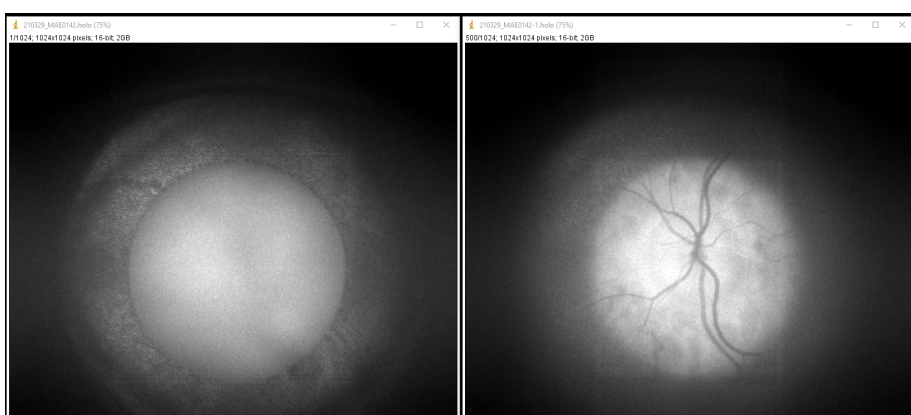
open. Its aspect ratio is 2:1 because the original interferograms were recorded with an anamorphic frame with the same aspect ratio, and the reconstruction was done on the same grid size..



You can resize the frame by clicking on “Image/Adjust/Size...” and setting equal width and height values.



The displayed frame should show reconstructed holographic images of the anterior and the posterior segments of a human eye.



Congratulations now you know how to open .holo files with ImageJ !

holo file writer

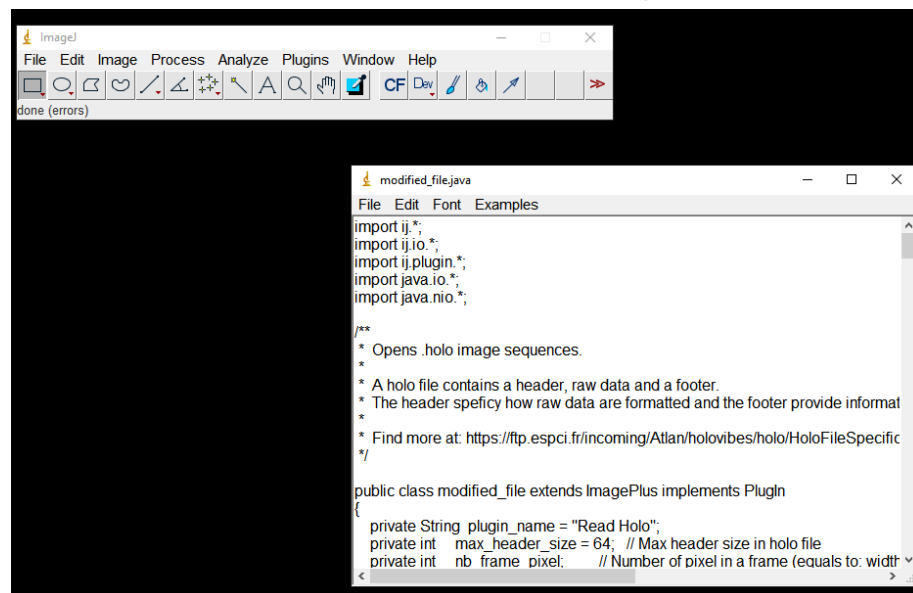
You can also save .holo files. Click on the “Save Holo” plugin in ImageJ while a file is open in ImageJ. Choose a name for this new .holo file and save it.

To check if this new file is a .holo file you can use the “Open Holo” plugin and open it.

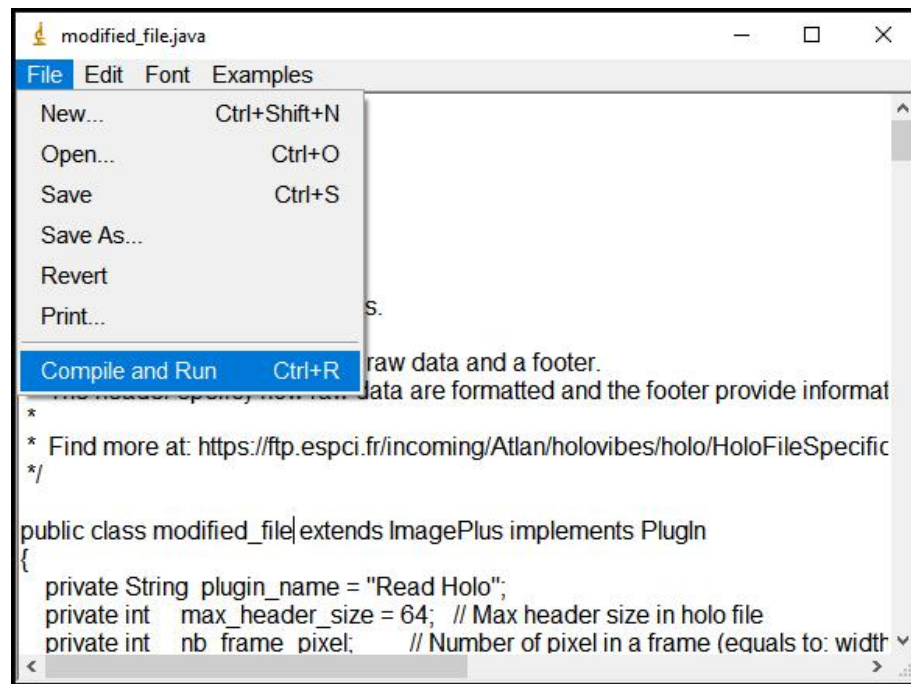
Congratulations now you know how to save .holo files with ImageJ !

Compiling .java files to .class to customize the plugins

If you want to make some modifications in these plugins you can do it! For that, you can customize the .java files in the “Holo” folder with the text editor of your choice. After that, open the “modified_file.java” file with ImageJ (Ctrl + O).



Compile and Run this new file (Ctrl + R).



You can observe in the “Holo” folder a new “modified_file.class” file. This is the executable file.

Restart ImageJ if it was open. Now you can use your customized plugin with ImageJ.

