

# Step-by-step protocol for data acquisition and reconstruction of holographic images in high-throughput single-pixel holography

**DaiXuan Wu**

Key Laboratory of Optoelectronic Materials and Technologies, School of Electronics and Information Technology, Sun Yat-sen University <https://orcid.org/0000-0002-9232-3442>

**Jiawei Luo**

Key Laboratory of Optoelectronic Materials and Technologies, School of Electronics and Information Technology, Sun Yat-sen University

**Guoqiang Huang**

Key Laboratory of Optoelectronic Materials and Technologies, School of Electronics and Information Technology, Sun Yat-sen University

**Yuanhua Feng**

Department of Electronic Engineering, College of Information Science and Technology, Jinan University

**Xiaohua Feng**

Department of Bioengineering, University of California

**Runsen Zhang**

Institute of Photonics Technology, Jinan University

**Yuecheng Shen** (✉ [shenyuecheng@mail.sysu.edu.cn](mailto:shenyuecheng@mail.sysu.edu.cn))

Key Laboratory of Optoelectronic Materials and Technologies, School of Electronics and Information Technology, Sun Yat-sen University

**Zhaohui Li** (✉ [lzh88@mail.sysu.edu.cn](mailto:lzh88@mail.sysu.edu.cn))

Key Laboratory of Optoelectronic Materials and Technologies, School of Electronics and Information Technology, Sun Yat-sen University

---

## Method Article

**Keywords:** single-pixel holography, biological imaging

**Posted Date:** August 12th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.pex-1576/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

# Abstract

With the strong desire to develop an efficacious imaging modality that can scrutinize complex-valued objects with intricate structures, such as a myriad of biological samples, we develop high-throughput single-pixel holography (SPH). The protocol aims to describe the detailed procedures of this work.

## Introduction

The process to reconstruct a single-pixel holographic image can be briefly described into three parts. (1) Projection of modulated light patterns onto the complex-valued objects to be measured; (2) Data acquisition of transmitted light in a holographic system; (3) Reconstruction algorithm realized using customized coded for holographic reconstruction. All three parts have their respective functionalities, which will be introduced in Procedure section in order.

## Reagents

## Equipment

Statement of system requirements:

1. All the code demonstrations and experimental performances can be successfully executed using the MATLAB v2016a (including the later versions) and Visual Studio 2013 (including the later versions).
2. In this work, we chose to use the MATLAB v2020b and the Visual Studio 2019 community version. The operating system is Windows 10.
3. During the data acquisition, the physical presence of the digital micromirror device (DMD) and the data acquisition card (DAC) is required to execute respective C++ files successfully. Both the DMD and the DAC are the required non-standard hardware for the experiment.

Commercialized software, including the MATLAB, the Visual Studio, the software development kits (SDKs) of the DMD from Vialux, and the DAC from NI, can be accessed from official websites, as well as their installation guides. For our computer (Intel(R) Core(TM) i5-8600K CPU@3.60GHz), it generally takes about one hour of installation time on a desktop personal computer, respectively.

## Procedure

### Preparation of the modulation patterns

This part aims to produce the binary data to be preloaded into the DMD as modulation patterns. The file 'DMD\_hadamard256.m' in this file folder is used to generate Hadamard-like patterns with orders up to  $256 \times 256$ . Hadamard-like patterns with different orders can be generated as well by adjusting the parameter 'number' in this file. The execution of this file with commercialized software MATLAB generates a

sequence of Hadamard-like patterns in the format of .txt binary files. To minimize phase drifting caused by environmental disturbance during the experiment, the first order of Hadamard-like patterns (DC signal) is inserted after every 16 modulation patterns as the tracking base. Therefore, 73,728 patterns are generated in a sequence and it takes about 25 minutes to finish this one-time preparation. This MATLAB code is provided in 'Supplementary Code', which can be accessed in the public repository Zenodo [1].

A DMD (Texas Instrument DLP7000&DLPC410) is employed to project Hadamard-like patterns to the sample. The software and the control board of the DMD are provided by Vialux company (model number Vialux V7001). The Hadamard-like patterns in the format of .txt binary files can be uploaded to the random-access memory (RAM) in advance. Unfortunately, we do not provide a series of .txt files of Hadamard-like patterns for saving space as it occupies a large memory of about 50 Gbit (less than the 64 Gbit of the RAM of the DMD). After execution of this MATLAB file, the regenerated .txt binary files will be collected in the default document 'data\_image256\_3\_3\_8bitplane\_russia\_persuitbasis2\_txt'.

## **Data acquisition**

This part describes how to collect the data. The main C++ code file that is implemented in the Visual Studio C++ project is included in the main folder named 'joint\_256bitplane8', which is provided in 'Supplementary Code' that can be accessed in the public repository Zenodo [1].

This folder includes some necessary files of the Visual Studio C++ project to control both the DMD and the DAC. In this project, the execution of an integrated C++ source file named 'joint\_256bitplane8.cpp' enables the DMD to display the desired Hadamard-like patterns with full speed and the DAC to acquire data subsequently. After preloading all the desired Hadamard-like patterns onto the DMD, it takes about 3 seconds to finish the data acquisition once. Please visit the website

<https://visualstudio.microsoft.com/zh-hans/> to download the Visual Studio 2019 community version.

However, the successful execution of these C++ files requires the correct configuration of environment variables in a "normal" desktop computer as well as the physical presence of the DMD and the DAC. It means that the data acquisition step cannot be run successfully without any supporting SDKs from Vialux and NI that manage the DMD and the DAC.

To gain control of the DMD and the DAC respectively, please visit the website

<https://www.vialux.de/en/download.html> to download corresponding control files for the DMD and

<https://www.ni.com/zh-cn/support/downloads/drivers/download.ni-daqmx.html#288248> for the DAC.

## **Holographic reconstruction**

This part illustrates how to use MATLAB to reconstruct holographic images from the experimentally measured data. In the file folder named 'holographic reconstruction' provided in 'Supplementary Code' that can be accessed in the public repository Zenodo [1], there are two subfolders named 'stained tissue from mouse tails' and 'unstained tissue from mouse brains', corresponding to different types of biological tissue. In each folder, the MATLAB file named 'singlepixel\_dataprocessing\_complex.m' is the main file executed for holographic reconstruction. After executing the main file and loading the converted MATLAB data named 'biological\_sample.mat' and 'background\_curvature.mat', output files including many intermediate variables are stored in the folder named 'results'. Among them, the expected amplitude and phase images are named 'imaging-amplitude.fig' and 'imaging-phase.fig', respectively.

A flowchart of the step-by-step procedures to reconstruct holographic images is shown in Fig. 1.

## Troubleshooting

## Time Taken

## Anticipated Results

## References

1. Daixuan Wu. (2021, July 6). Supplementary Code for "Imaging biological tissue with high-throughput single-pixel compressive holography". Zenodo. <http://doi.org/10.5281/zenodo.5078499>

## Figures

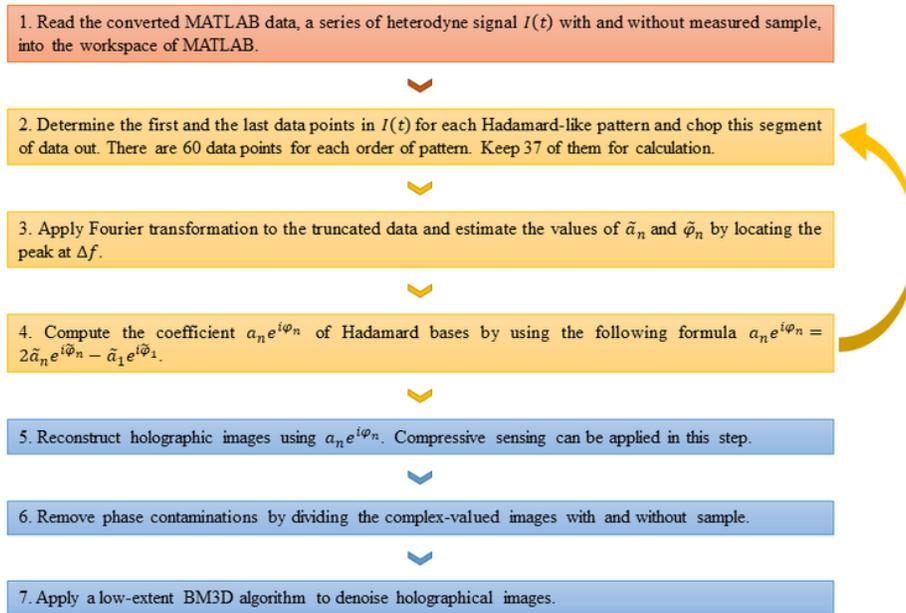


Figure 1

A flowchart of the reconstruction process