Assignment Five: Stereo-Vision with Phase Shifting Structure Light

David Gu

Computer Science Department
Stony Brook University

gu@cs.stonybrook.edu

August 5, 2022

Stereo-vision with Structured Light

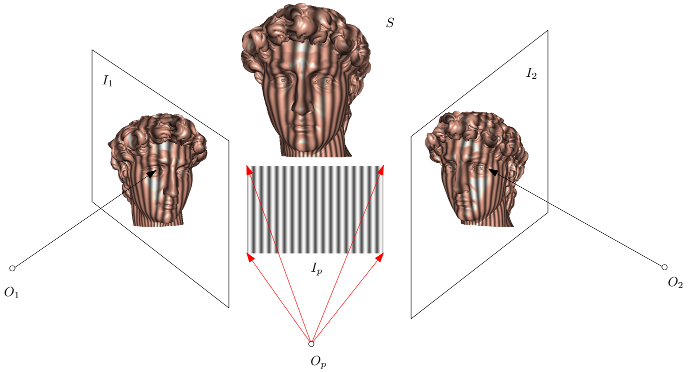
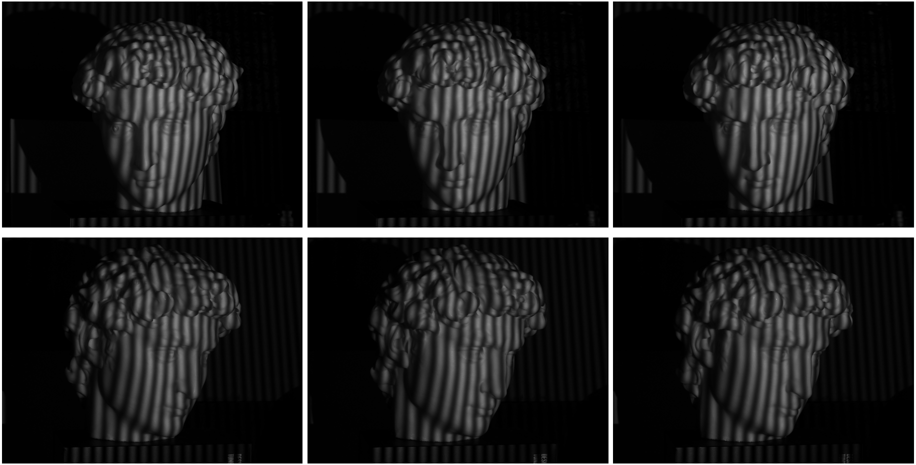


Figure: The fringe pattern for the digital projector or the LCD display. The left and right camera optical centers and image planes are (O_1, I_1) and (O_2, I_2) respectively. The projector optical center and image plane are (O_p, I_p).

Fringe Images



(a). $I_1(x, y)$ (b). $I_2(x, y)$ (c). $I_3(x, y)$

Figure: Fringe images: the top row shows the images captured by the left camera, the bottom row show shows those by the right camera.

Image Decomposition

The image intensity is formulated as

$$\begin{aligned} I_1(x,y) &= I'(x,y) + I''(x,y)\cos(\Phi(x,y) - 2\pi/3) \\ I_2(x,y) &= I'(x,y) + I''(x,y)\cos(\Phi(x,y)) \\ I_3(x,y) &= I'(x,y) + I''(x,y)\cos(\Phi(x,y) + 2\pi/3) \end{aligned} \quad (1)$$

where $I'(x,y)$ is the ambient, $I''(x,y)$ the intensity modulation, $\Phi(x,y)$ the absolute phase.

Image Decomposition

The ambient, the modulation and the relative phase can be obtained by

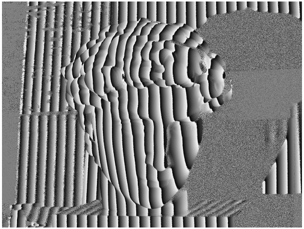
$$
\begin{aligned}
I'(x,y) &= \frac{1}{3}[I_1(x,y) + I_2(x,y) + I_3(x,y)] \\
I''(x,y) &= \frac{1}{3}\sqrt{3(I_1 - I_3)^2 + (2I_2 - I_1 - I_3)^2} \\
\varphi(x,y) &= \tan^{-1}\frac{\sqrt{3}(I_1 - I_3)}{2I_2 - I_1 - I_3}
\end{aligned}
\tag{2}
$$

where the relative phase $\varphi(x,y)$ is from $-\pi$ to π,

$$\varphi(x,y) = \Phi(x,y) \mod 2\pi.$$

ambient $I'(x, y)$

modulation $I''(x, y)$



wrapped phase $\varphi(x, y)$

Figure: The ambient, modulation and wrapped phase computed from the fringe images in Fig. 2.

Phase Unwrapping

Problem (Phase Unwrapping)

Given an image $\mathcal{I}$ of m rows and n columns, each pixel position is represented as a pair of indices (i,j), where $1 \leq i \leq m$, $1 \leq j \leq n$. We use $p \in [1,m] \times [1,n]$ to represent a point in the image plane. The wrapped phase at the pixel p is denoted as φ_p, the wrap uncount at p as k_p, the unwrapped phase Φ_p, then

$$\Phi_p = \varphi_p + 2\pi k_p, \quad \forall p \in \mathcal{I}. \tag{3}$$

The wrap count function $k : \mathcal{I} \to \mathbb{Z}$ is the unkown function.

Phase Unwrapping - Multiple Wavelength Method

One can use multiple projection fringe pattern with different wavelengths to recover the absolute phase. The wavelength λ_i's satisfy the relation

$$\lambda_i = 2\lambda_{i+1}, \quad i = 0, 1, 2, \dots, n.$$

λ_0 is big enough to cover the whole scanning range, so the relative phase φ_0 equals the absolute phase Φ_0, $\Phi_0 \leftarrow \varphi_0$. By the relation

$$\Phi_i(p)\lambda_i = \Phi_{i+1}(p)\lambda_{i+1} \implies 2\Phi_i = \Phi_{i+1}$$

We obtain

$$\Phi_{i+1} - \varphi_{i+1} = 2\Phi_i - \varphi_{i+1} \implies k_{i+1} = \text{round}\left(\frac{\Phi_i}{\pi} - \frac{\varphi_{i+1}}{2\pi}\right)$$

and

$$\Phi_{i+1} = 2\pi k_{i+1} + \varphi_{i+1}$$

Multiple Wavelength

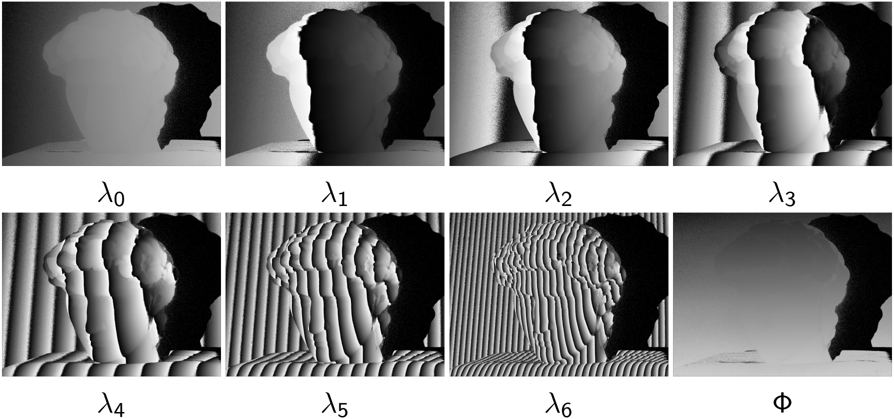


Figure: Multiple wavelength method.

Stereo-Matching

- The projector pixel coordinates are encoded by horizontal and vertical phases (Φ_x, Φ_y)
- Each pixel $p \in \mathcal{L}$ on the left image is lit by a projector pixel, encoded by the horizontal and vertical unwrapped phase $\Phi_x(p)$ and $\Phi_y(p)$;
- Each pixel $q \in \mathcal{R}$ on the right image is lit by a projector pixel, encoded by the horizontal and vertical unwrapped phase $\Phi_x(q)$ and $\Phi_y(q)$;
- Match each pixel on the left image $p \in \mathcal{L}$ to the best pixel $q \in \mathcal{R}$,

$$\text{argmin}_{q\in\mathcal{R}} \|\Phi_x(p) - \Phi_x(q)\|^2 + \|\Phi_x(q) - \Phi_x(q)\|^2$$

- In order to speed up the stereo-matching, the image pixels are mapped to the projector pixels, namely each projector pixel is associated with a cluster pixels on the left image and another cluster of pixels on the right image.

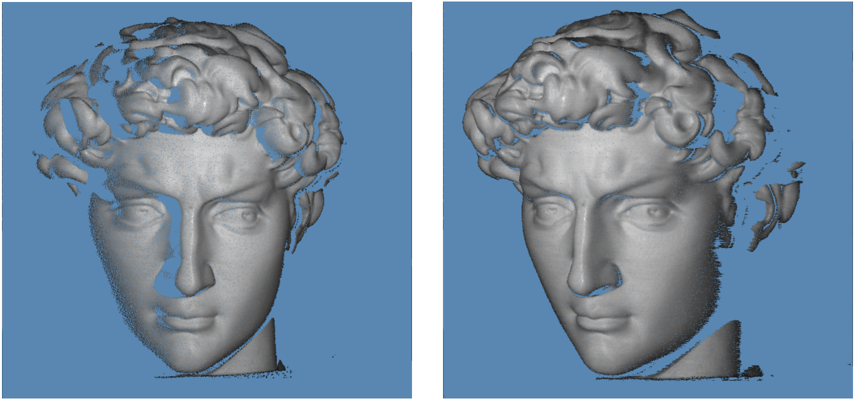


Figure: Reconstructed Point cloud.

Instruction

Dependencies

In the 3rdparty directory:

1. ‘MeshLib‘, a mesh library based on halfedge data structure.
2. ’eigen’, sparselinear solver library.
3. ‘ScanLib‘, library for 3D scanning.

Directory Structure

Source files:

- 3rdparty/ScanLib/phase, headers, source files for phase shifting structured light;
- 3rdparty/ScanLib/Stereo, headers, source files for stereo-matching;
- 3rdparty/ScanLib/LightField, headers, source files for camera calibration;
- CMakeLists.txt, CMake configuration file;

Directory Structure

Data files:

- data/17, 3×14 raw images captured by the left camera;
- data/33, 3×14 raw images captured by the right camera;
- data/17_33, intermediate computational results;
- data/configure, two camera calibration files;
- data/scripts, the script files.

File Structure

Script files:

- data/scripts/configuration.text, configuration file for processing;
- data/scripts/HorizontalVerticalProcess.bat, master script file;
- data/scripts/17_step_1_horizontal_vertical_phase_unwrap.bat, phase unwrap for the left camera images;
- data/scripts/33_step_1_horizontal_vertical_phase_unwrap.bat, phase unwrap for the right camera images;
- data/scripts/HorizontalVerticalStereoMatch.bat, stereo-matching script;
- data/scripts/ViewMesh.bat, view reconstructed point cloud;

Configuration

Before you start, read README.md carefully, then go three the following procedures, step by step.

1. Install [CMake](https://cmake.org/download/).
2. Download the source code of the C++ framework.
3. Configure and generate the project for Visual Studio.
4. Open the .sln using Visual Studio, and complie the solution.
5. Finish your code in your IDE.
6. Run the executable program.

3. Configure and generate the project

1. open a command window
2. cd PhaseUnwrap_Skeleton_0805_2022,
3. mkdir build
4. cd build
5. start CMake GUI, configure
6. Specifiy OpenCV_Dir
7. configure, generate,
8. Open Project

5. Finish your code in your IDE

- You need to modify the file: HorizontalVerticalMultipleWaveLengthPhseUnwrapper.cpp
- search for comments "insert your code"
- Modify the functions:

$$compute\_modulation();$$
$$compute\_ambient();$$
$$compute\_phase();$$

5. Finish your code in your IDE

- You need to modify the file: CapturePhaseUnwrapper.cpp
- search for comments "insert your code"
- Modify the function:

  $int\_double\_wavelength\_phase\_unwrap(UnwrappedPhase, WrappedPhase)$

  The input parameters are

  1. the unwrapped phase with the wavelength λ_1;
  2. the wrapped phase with the wavelength $\lambda_2 = 1/2\lambda_1$
  3. the return value is the wrap count.

5. Finish your code in your IDE

- You need to modify the file: HorizontalVerticalStereoMatcher.cpp
- search for comments "insert your code"
- Modify the function:

$$
\begin{aligned}
\_match\_one\_camera\_pixel(&intleft_i, intleft_j, \\
&intu, intv, \\
&intdisparity_threshold);
\end{aligned}
$$

The input parameters are

1. the pixel indices on the left image (left_i,left_j);
2. the corresponding projector pixel (u, v);
3. the disparity_threshold;

Find the pixel on the right image, such that the absolute phase difference is minimized.

6. Run the executable program (Demo)

Testing Procedure

1. Replace phaseunwrap.exe in the directory data/bin/;
2. Remove all the intermediate computational results in the folder data/17 and data/33;
3. Double click on the script data/scripts/HorizontalVerticalProcess.bat;

Many image windows will popup, press any key to continue, until the point cloud window pops up.