Report

Numerical Methods in Optical Techniques

Project part 3: analysis

Author: Mateusz Surma

1. Aim

Goal of this project was to produce algorithm allowing to obtain basic properties of micro-lenses array based on 5 holograms. Procedure of preparing the project was divided in 3 parts. This report will concentrate on the last part.

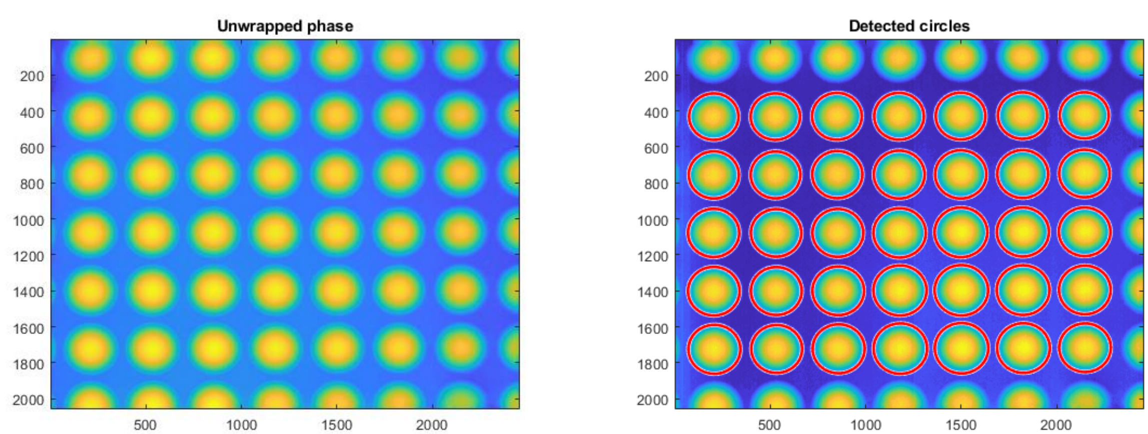
2. Summary of previous tasks

First part focused on reconstruction of the phase from temporally shifted frames, preparation of the propagation algorithm (angular spectrum method) and development of autofocusing algorithm. Thanks to all aforementioned steps phase and intensity distribution of micro-lens array (in its plane) was obtained. Resulting phase distribution was unwrapped and passed to second part of the project.

In the second part of the project detection of lenses was realized. This part was realized with c++. After series of procedures including Gaussian convolution (blurring), binarization, Laplacian convolution (edge detection) and finally circle detection with Hough transform. Result of this algorithm was file describing best fitting circles (position of the centre and radius) that contained lenses. In the final version of the project executable file performing this algorithm is called from Matlab script providing full automation of the process (achieved by means of system command).

3. Description of algorithm and results from part 3

The last part was concentrated on limitation of selected for analysis lenses and calculation of the parameters describing them. All detected circles are marked on the Figure 1 together with unwrapped phase.



a) b)

Figure 1 a) Unwrapped phase and b) all detected circles

In order to be sure that detected circles will contain whole lenses thy have to be carefully selected. In order to remove from list of detected objects those which are on the edges simple algorithm based on the position of centres can be devised:

```
%% find and remove circles that are on the edges
edgePercent = 15;
%find elements that are in the vicinity of edges
removeX = find(save(:,1) <= size(pha,2)*edgePercent/100 |  save(:,1) >=
size(pha,2)*(100-edgePercent)/100);
removeY = find(save(:,2) <= size(pha,1)*edgePercent/100 |  save(:,2) >=
size(pha,1)*(100-edgePercent)/100);
%remove selected circles
save(removeX, :) = [];
save(removeY, :) = [];
figure; imagesc(unph);
viscircles(save(:,1:2), save(:,3)); title('After removing edge elements');
```

Distance from the edge from which no circles will be taken for further analysis is kept as percentage in the variable `edgePercent`. Result of this part of the code can be seen on the Figure 2. For succeeding calculations 25 detected lenses were used.

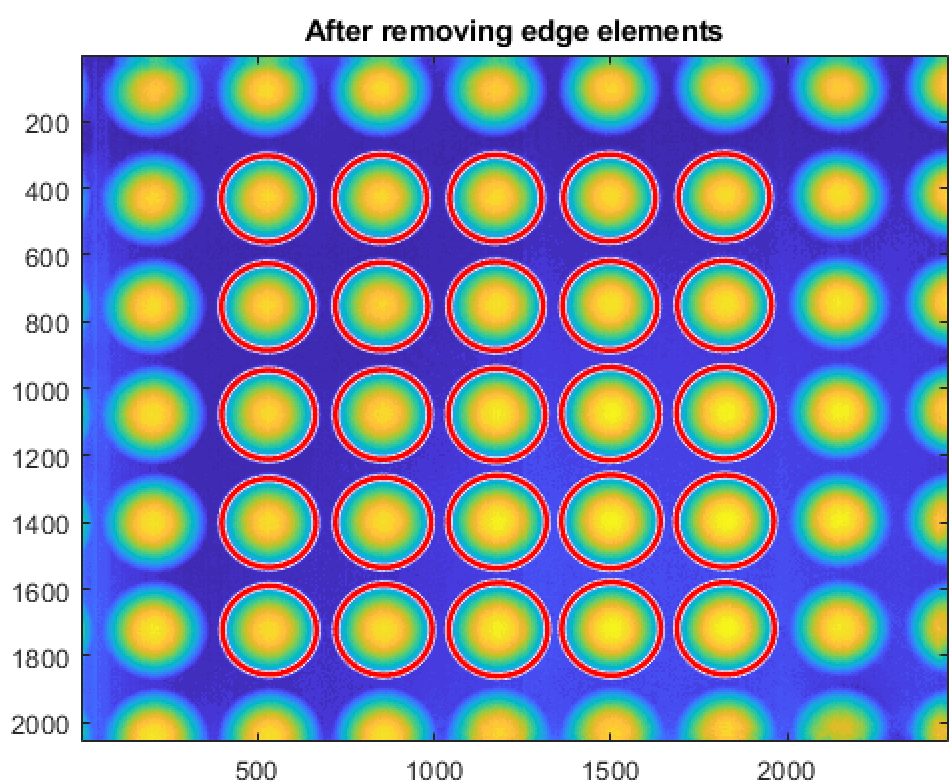


Figure 2 Circles after removing elements close to the edges

Further procedure was concerning the calculation of the parameters. Some of the most important parameters of micro-lenses arrays are: diameter (d), radius of curvature (ROC), numerical aperture (NA), height (h) and focal length (f). All of aforementioned where calculated together with standard deviations and depicted in Table 1.

Table 1 Calculated values with standard deviation and nominal values (provided or calculated)

Parameter	Unit	Value	Standard Deviation	Nominal values
d	um	92.65	2.55	95
h	um	9.67	0.32	9.80
ROC	um	115.84	3.63	120
f	um	253.47	7.95	262.58
NA	-	0.21	0.004	0.19

Calculation procedures for micro-lens' parameters have following shape:

a. Diameter – has double the value of detected radius

```
r_mat = save(:,3)*ds; % temporary variable storing all radius'
r = mean(r_mat);
r_std = std(r_mat);
d = 2*r;
d_std = 2* r_std;
```

Matrix save contains data of selected circles. Final value is calculated as mean from all radii. Standard deviation is calculated. Similar situation is happening in further calculations so it will not be depicted in latter parts of report.

b. Height – in this part it was interpreted as so called lens sag (part of lens standing out of the plate). All height were calculated from following equation:

$$\varphi = \frac{2\pi(n_s - n_0)h}{\lambda} \leftrightarrow h = \frac{\lambda\varphi}{2\pi(n_s - n_0)},$$

where φ – phase change, λ – wavelength of light used to obtain the hologram (here 0.6328; in code *lamb*), n_s - refractive index of array's material (here 1.45701), n_0 - refractive index of environment (here 0). In order to obtain only value of sag from height at the centre of the lens was subtracted height at the edge (calculated as average of 4 points from lenses edge).

```
h_mat = zeros(size(save(:,3))); % vector storing final values of height
tmp_h_part = lamb/(2*pi*(ns-n0)); % repeating constant part of calculation
% calculations done for each radius
for i = 1:size(save(:,3),1)
    tmp_r = save(i,3);
    h_cent = double(unph(save(i,2),save(i,1)))*tmp_h_part;
    h_edge1 = double(unph(save(i,2)-tmp_r,save(i,1)))*tmp_h_part;
    h_edge2 = double(unph(save(i,2)+tmp_r,save(i,1)))*tmp_h_part;
    h_edge3 = double(unph(save(i,2),save(i,1)+tmp_r))*tmp_h_part;
    h_edge4 = double(unph(save(i,2),save(i,1)-tmp_r))*tmp_h_part;
    h_edge = mean([h_edge1, h_edge2, h_edge3, h_edge4]);
    h_mat(i) = h_cent; - h_edge;
end
```

Variable `unph` contains unwrapped phase.

c. ROC – was calculated with following formula:

$$ROC = \frac{h(K+1)}{2} + \frac{r^2}{2h},$$

where K – conic constant (for spherical lens 0), r – detected radius.

d. Focal length – was obtained with formula:

$$f = \frac{ROC}{n_s - 1}.$$

e. NA – was calculated with following equation:

$$NA \approx \frac{h}{r}.$$

Additionally it has to be mentioned that all obtained NA has been used.

In provided Matlab script simple method of background removal was used but since change of results of circle detection after using it was not significant it will not be described in detail in this report.

4. Conclusions

Analysing obtained values with nominal (Table 1) it clearly visible that calculated parameters are close to nominal. However, detected lenses can differ between each other which is shown by standard deviation. It should be also noted that methods of calculation and detection are not free of errors. Approximation are used in the calculations (for example equation for numerical aperture). Detection of circles is also prone to errors.