Supporting Information

Complex Inverse Design of Meta-Optics by Segmented Hierarchical Evolutionary Algorithm

Zhongwei Jin[1⊥], Shengtao Mei[1⊥], Shuqing Chen[2⊥], Ying Li[2⊥], Chen Zhang[3], Yanliang He[2], Xia Yu[1], Changyuan Yu[1,4], Joel K. W. Yang[5,6], Boris Luk'yanchuk[7,8], Shumin Xiao[3*], Cheng-Wei Qiu[1, 9*]

[1]Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117583, Singapore

[2]International Collaborative Laboratory of 2D Materials for Optoelectronics Science and Technology, Shenzhen University, Nanhai Ave 3688,Shenzhen,Guangdong,P.R.China,518060

[3]State Key Laboratory on Tunable Laser Technology, Ministry of Industry and Information Technology Key Lab of Micro-Nano Optoelectronic Information System, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, Guangdong 518055, P.R. China

[4]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

[5]Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372

[6]Institute of Materials Research and Engineering, A*STAR (Agency for Science, Technology and Research), 2 Fusionopolis Way, #08-03 Innovis, 138634 Singapore

[7]Division of Physics and Applied Physics, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371, Singapore.

[8]Faculty of Physics, Lomonosov Moscow State University, Moscow 119991, Russia.

[9]SZU-NUS Collaborative Innovation Center for Optoelectronic Science and Technology, Shenzhen University, Shenzhen 518060, People's Republic of China

[⊥]These authors contributed equally to this work.

[*]Corresponding authors: *chengwei.qiu@nus.edu.sg; shumin.xiao@hit.edu.cn*

Part 1. Implementation of SHEA (segmented hierarchical evolutionary algorithm) for the Broadband Si Hologram

Step 1: Reorganizing the Optimization Task to a Segmentation Hierarchy Task & Evaluating Error Value based on Initial Genome$_L$

Since the original Genome pixel array size should be $M \times M (M = 302)$, here in the lower segmentation, we set ***Genome$_L$*** have $N \times N$ (we chose N to be 151 in this case) super-pixels, each super-pixel contains $n \times n\ (n = 2)$ pixels. While the target observation plane is set as the same size with the metasurface but divided into $(2N + 1) \times (2N + 1)$ pixels for higher accuracy. Before launching the optimization process, we first calculated the complex electric fields from every nanopillar of the metasurface to each pixel of the target plane. And the complex electric fields from each nanopillar to each pixel on the target plane contains x, y and z components. Therefore, in this case, the size of the initial complex electric fields in the lower segmentation can be estimated as: $3 \times (151 \times 151 \times 303 \times 303) \times 8/(1024)^3 \approx$ 46.8GB, while the size of the initial complex electric fields in the higher segmentation can be estimated as: $3 \times (302 \times 302 \times 605 \times 605) \times 8/(1024)^3 \approx 746.2$GB, both will exceed the RAM of a personal computer. However, due to the symmetric property of the light field penetrating through the Si nanopillar according to the Huygens-Fresnel principle, only the electrical fields of the center nanopillar on the metasurface to the target plane need to be calculated and saved in a matrix, while the complex electrical fields coming from all the other nanopillars can be obtained from the matrix through translation operation by comparing the relative position of the nanopillar with the center one. Therefore, in the lower segmentation, the size of the initial complex electric fields in the lower segmentation can be reduced to: $3 \times \big((2 \times 303 - 1) \times (2 \times 303 - 1)\big) \times 8/(1024)^3 \approx 8.4$ MB, and the size of the initial complex electric fields in the higher segmentation can be estimated as: $3 \times \big((2 \times 605 - 1) \times (2 \times 605 - 1)\big) \times 8/(1024)^3 \approx 33.5$ MB. Here, the initial matrices to save the complex fields of the center super-pixel on metasurface to all the pixels on the observation plane in the lower segmentation are denoted as $Efield_{1x}$, $Efield_{1y}$ and $Efield_{1z}$ for x, y and z components respectively. The intensity distribution of the target image is saved as $Target_1$.

Code Section 1

$Efield_{1x} = zeros(2(2N + 1) - 1, 2(2N + 1) - 1);$

$Efield_{1y} = zeros(2(2N + 1) - 1,2(2N + 1) - 1);$

$Efield_{1z} = zeros(2(2N + 1) - 1,2(2N + 1) - 1);$

$a = 2$;

for $ii : 1 \rightarrow 2(2N+1)-1$ **do**

for $jj : 1 \rightarrow 2(2N+1)-1$ **do**

Calculate the complex fields of the center super-pixel on metasurface to the pixel (*ii, jj*) on the observation plane according to point-source model and saved in $Efield_{1x}(ii,jj)$, $Efield_{1y}(ii,jj)$ and $Efield_{1z}(ii,jj)$ respectively.

end

end

Randomly generate an $N \times N$ matrix ***Genome*$_L$** which only contains "0" or "1";

$Efield\_L_x = zeros(2N+1, 2N+1)$;

$Efield\_L_y = zeros(2N+1, 2N+1)$;

$Efield\_L_z = zeros(2N+1, 2N+1)$;

for $ii : 1 \rightarrow N$ **do**

for $jj : 1 \rightarrow N$ **do**

if ***Genome*$_L$** $(ii, jj) = 1$ **do**

$mx = ii - (N+1)/2$;

$my = jj - (N+1)/2$;

$Efield\_L_x = Efield\_L_x + Efield_{1x}\big((N+1-a*mx):(3N+1-a*mx),$
$(N+1-a*my):(3N+1-a*my)\big)$;

$Efield\_L_y = Efield\_L_y + Efield_{1y}\big((N+1-a*mx):(3N+1-a*mx),$
$(N+1-a*my):(3N+1-a*my)\big)$;

$Efield\_L_z = Efield\_L_z + Efield_{1z}\big((N+1-a*mx):(3N+1-a*mx),$
$(N+1-a*my):(3N+1-a*my)\big)$;

$Target_1 = Efield\_L_x .* conj(Efield\_L_x) + Efield\_L_y .* conj\big(Efield\_L_y\big) + Efield\_L_z .*$
$conj(Efield\_L_z)$;

end

end

end

Evaluate the value of the fitness function *F_L* based on $Target_1$.

Step 2: Start Iterative Optimization in the Lower Segmentation until Reaching the Converging Conditions of the Lower Segmentation

The converging condition in the lower segmentation is determined as when the value of the fitness function F_L stays unchanged (considering accuracy in 3 decimal places of F_L) for three rounds.

Code Section 2

$Count\_L = 0; Num\_Generation\_L = 0;$

While 1

Randomly generate a $1 \times N^2$ mutation matrix $Mutation1$ where integers from 1 to N^2 are arranged in the matrix in a random sequence;

for $jj: 1 \rightarrow N^2$ **do**

$MutationX1 = mod(Mutation1(1, jj), N) + 1;$

$MutationY1 = ceil(Mutation1(1, jj)/N);$

$nx = MutationX1 - (N + 1)/2;$

$ny = MutationY1 - (N + 1)/2;$

$New\_Efield\_L_x = Efield\_L_x + (-1)^{Genome_L(MutationX1, MutationY1)} * Efield_{1x}\big((N + 1 - a * nx): (3N + 1 - a * nx),\ (N + 1 - a * ny): (3N + 1 - a * ny)\big);$

$New\_Efield\_L_y = Efield\_L_y + (-1)^{Genome_L(MutationX1, MutationY1)} * Efield_{1y}\big((N + 1 - a * nx): (3N + 1 - a * nx),\ (N + 1 - a * ny): (3N + 1 - a * ny)\big);$

$New\_Efield\_L_z = Efield\_L_z + (-1)^{Genome_L(MutationX1, MutationY1)} * Efield_{1z}\big((N + 1 - a * nx): (3N + 1 - a * nx),\ (N + 1 - a * ny): (3N + 1 - a * ny)\big);$

$New\_Target_1 = New\_Efield\_L_x .* conj(New\_Efield\_L_x) + New\_Efield\_L_y .* conj\big(New\_Efield\_L_y\big) + New\_Efield\_L_z .* conj(New\_Efield\_L_z);$

Calculate New_F_L according to *New_*$Target_1$ *;*

If $New\_F\_L > F\_L$ **do**

$New\_Genome_L = Genome_L$

If $Genome_L(MutationX1, MutationY1) == 1$ **do**

$New\_Genome_L(MutationX1, MutationY1) = 0;$

else $New\_Genome_L(MutationX1, MutationY1) = 1;$

end

$Genome_L = New\_Genome_L$;

$Efield\_L_x = New\_Efield\_L_x$;

$Efield\_L_y = New\_Efield\_L_y$;

$Efield\_L_z = New\_Efield\_L_z$;

$F\_L = New\_F\_L$;

end

$Num\_Generation\_L = Num\_Generation\_L + 1$;

If $F\_L(Num\_Generation\_L) == F\_L(Num\_Generation\_L - 1)$

$Count\_L = Count\_L + 1$;

else $Count\_L = 0$

end

If $Count\_L == 3$

break;

end

end

Step 3 Generate Initial Genome$_H$ in the Higher Segmentation Based on the Final Evolutionary Result of Genome$_L$ from the Lower Segmentation

In the higher segmentation, the pixel size on the metasurface is $M \times M$. While the target observation plane is set as the same size with the metasurface but divided into $(2M + 1) \times (2M + 1)$ pixels for higher accuracy and the intensity distribution of the target image is saved as $Target_2$.

Code Section 3

$Genome_H = zeros(M, M)$;

for $ii : 1 \rightarrow N$ **do**

for $jj : 1 \rightarrow N$ **do**

$Genome_H(2(ii-1)+1, 2(jj-1)+1) = Genome_H(2(ii-1)+1, 2(jj-1)+2) =$

$Genome_H(2(ii-1)+2, 2(jj-1)+1) = Genome_H(2(ii-1)+2, 2(jj-1)+2) =$

$Genome_L(ii, jj)$;

end

end

$Efield_{2x} = zeros(2(2M+1)-1, 2(2M+1)-1);$

$Efield_{2y} = zeros(2(2M+1)-1, 2(2M+1)-1);$

$Efield_{2z} = zeros(2(2M+1)-1, 2(2M+1)-1);$

for $ii : 1 \rightarrow 2(2M+1)-1$ **do**

for $jj : 1 \rightarrow 2(2M+1)-1$ **do**

Calculate the complex fields of the center pixel on metasurface to the pixel (*ii, jj*) on the observation plane according to point-source model and saved in $Efield_{2x}(ii, jj)$, $Efield_{2y}(ii, jj)$ and $Efield_{2z}(ii, jj)$ respectively.

end

end

$Efield\_H_x = zeros(2M+1, 2M+1);$

$Efield\_H_y = zeros(2M+1, 2M+1);$

$Efield\_H_z = zeros(2M+1, 2M+1);$

for $ii : 1 \rightarrow M$ **do**

for $jj : 1 \rightarrow M$ **do**

if $Genome_H(ii, jj) = 1$ **do**

$mx = ii - (M+1)/2;$

$my = jj - (M+1)/2;$

$Efield\_H_x = Efield\_H_x + Efield_{2x}\big((M+1-a*mx):(3M+1-a*mx), (M+1-a*my):(3M+1-a*my)\big)$

$Efield\_H_y = Efield\_H_y + Efield_{2y}\big((M+1-a*mx):(3M+1-a*mx), (M+1-a*my):(3M+1-a*my)\big);$

$Efield\_H_z = Efield\_H_z + Efield_{2z}\big((M+1-a*mx):(3M+1-a*mx), (M+1-a*my):(3M+1-a*my)\big);$

$Target_2 = New\_Efield\_L_x .* conj(New\_Efield\_L_x) + New\_Efield\_L_y .* conj\big(New\_Efield\_L_y\big) + New\_Efield\_L_z .* conj(New\_Efield\_L_z);$

end

end

end

Evaluate the value of the fitness function F_H based on $Target_2$.

Step 4: Start Iterative Optimization in the Higher Segmentation until Reaching the Converging Conditions of the Higher Segmentation

The converging condition in the higher segmentation is determined as when the value of the fitness function F_H stays unchanged (considering accuracy in 3 decimal places of F_H) for three rounds.

Code Section 4

$Count\_H = 0; Num\_Generation\_H = 0;$

While 1

Randomly generate an $1 \times M^2$ mutation matrix $Mutation2$ where integers from 1 to M^2 are arranged in the matrix in a random sequence;

for $jj: 1 \rightarrow M^2$ **do**

$MutationX2 = mod(Mutation2(1, jj), M) + 1;$

$MutationY2 = ceil(Mutation2(1, jj)/M);$

$nx = MutationX2 - (M + 1)/2;$

$ny = MutationY2 - (M + 1)/2;$

$New\_Efield\_H_x = Efield\_H_x + (-1)^{Genome_H(MutationX2, MutationY2)} * Efield_{2x}\big((M + 1 - a * nx): (3M + 1 - a * nx),\ (M + 1 - a * ny): (3M + 1 - a * ny)\big);$

$New\_Efield\_H_y = Efield\_H_y + (-1)^{Genome_H(MutationX2, MutationY2)} * Efield_{2y}\big((M + 1 - a * nx): (3M + 1 - a * nx),\ (M + 1 - a * ny): (3M + 1 - a * ny)\big);$

$New\_Efield\_H_z = Efield\_H_z + (-1)^{Genome_H(MutationX2, MutationY2)} * Efield_{2z}\big((M + 1 - a * nx): (3M + 1 - a * nx),\ (M + 1 - a * ny): (3M + 1 - a * ny)\big);$

$New\_Target_2 = New\_Efield\_L_x .* conj(New\_Efield\_L_x) + New\_Efield\_L_y .* conj\big(New\_Efield\_L_y\big) + New\_Efield\_L_z .* conj(New\_Efield\_L_z);$

Calculate New_F_H according to New_Efield_H and New_Target_2 ;

If $New\_F\_H > F\_H$ **do**

$New\_Genome_H = Genome_H$

If $Genome_H(MutationX2, MutationY2) == 1$ **do**

$New\_Genome_H(MutationX2, MutationY2) = 0;$

else $New\_Genome_H(MutationX2, MutationY2) = 1;$

end

$Genome_H = New\_Genome_H;$

$Efield\_H_x = New\_Efield\_H_x;$

$Efield\_H_y = New\_Efield\_H_y;$

$Efield\_H_z = New\_Efield\_H_z;$

$F\_H = New\_F\_H;$

end

end

$Num\_Generation\_H = Num\_Generation\_H + 1;$

If $F\_L(Num\_Generation\_H) == F\_L(Num\_Generation\_H - 1)$

$Count\_H = Count\_H + 1;$

else $Count\_H = 0$

end

If $Count\_H == 3$

break;

end

end

Part 2. Discussion of Using Nanohole Structures for Full-Color Meta-Hologram

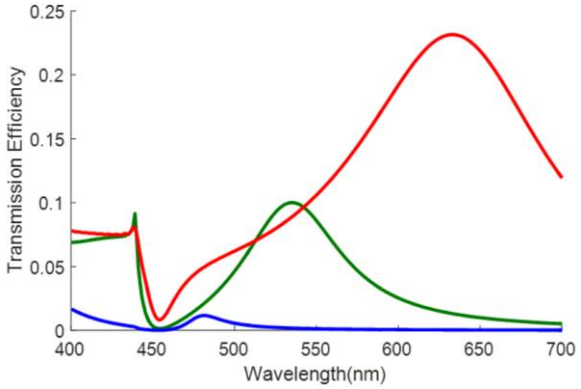

Figure S1. Transmission Efficiencies of Nanohole Structures Drilled on Aluminum Film. The red, green, and blue solid lines represent the transmission efficiencies of nanoholes optimized for 633nm, 532nm and 488nm respectively.

Table S1. Structural Dimensions and Transmission Efficiencies

	Width(nm)	*Length(nm)*	*Thickness(nm)*	*Transmission Efficiency (R)*	*Transmission Efficiency (G)*	*Transmission Efficiency (B)*
R	50	210	120	23.1%	8.74%	5.41%
G	50	160	120	1.3%	9.99%	2.81%
B	50	120	120	1.01%	0.188%	0.042%

In this section, we discuss how to use nanohole structures drilled on aluminum film to design full-color meta-holograms. Since the transmitted spectrum of nanohole structures relies primarily on the periods of the nanoholes, all the data provided in Figure S1 and Table S1 are the results from 5×5 arrays in order to maintain the periodicity of the nanoholes. This will decrease the resolution of the holographic image under the same-area pattern. Additionally, from Figure S1 and table S1, it can be seen that by using nanohole structures, we cannot directly decompose the color pictures into its three original component (Red, Green, and Blue), otherwise strong cross-talk between different colors will happen. Meanwhile, the transmission efficiencies for the blue light is extremely small, efficiency balance between different colors cannot be achieved from structural designs. Hence, we chose Si for our hologram devices in the main text.

Part 3. Experimental Setup for Measuring the Broadband Meta-hologram

Figure S2. Schematic of experimental setup for broadband Si hologram. HWP: half-wave plate; LP: linear polarizer; QWP: quarter-wave plate;

Part 4. Detailed Conversion Efficiencies of Si Nanopillars for Full-color Meta-hologram

Table S2. Conversion Efficiencies of Si Nanopillars in Different Super-cells

Super-cell Type	Conversion Efficiencies		
	445nm	532nm	633nm
R	0.32%	1.46%	31.46%
G	0.47%	31.13%	0.21%
B	22.17%	0.33%	2.98%
R+G	0.35%	18.47%	20.71%
R+B	19.1%	0.21%	27.64%
G+B	22.4%	27.19%	0.02%
R+G+B	17.27%	18.17%	16.98%

Part 5. Discussion on Applying SHEA for Phase-controlled Optical Devices

Our proposed SHEA can also be employed for designing phase-controlled optical devices as long as coupling with appropriate analytical models. For binary phase-controlled optical devices, we can use "0" and "1" to stand for meta-atoms with "0" and "π" phase change respectively. For *n*-step phase-controlled optical devices, we can use an *n*-digit binary code to represent the status of each pixel. For example, for an 8-step phase-controlled optical device, a 3-digit binary code can represent the status of each pixel ("000", "001", "010", "100", "011", "110", "101"and "111" to represent meta-atoms with phase change of "0", "π/4", "π/2", "3π/4", "π", "5π/4", "3π/2", "7π/4" respectively). For a *n*-step phase-controlled meta-hologram design, to update the phase change induced through each mutation, the major change of implementation in the lower segmentation during each mutation operation should be:

$$New\_Efield\_L_x = Efield\_L_x + \left(\exp\left(1i * \frac{\pi}{2^n}\right) - 1\right) * (-1)^{Genome_L(MutationX1, MutationY1)} *$$

$$Efield_{1x}\left((N + 1 - a * nx) : (3N + 1 - a * nx),\ (N + 1 - a * ny) : (3N + 1 - a * ny)\right);$$

$$New\_Efield\_L_y = Efield\_L_y + \left(\exp\left(1i * \frac{\pi}{2^n}\right) - 1\right) * (-1)^{Genome_L(MutationX1, MutationY1)} *$$

$$Efield_{1y}\left((N + 1 - a * nx) : (3N + 1 - a * nx),\ (N + 1 - a * ny) : (3N + 1 - a * ny)\right);$$

$$New\_Efield\_L_z = Efield\_L_z + \left(\exp\left(1i * \frac{\pi}{2^n}\right) - 1\right) * (-1)^{Genome_L(MutationX1, MutationY1)} *$$

$$Efield_{1z}\left((N + 1 - a * nx) : (3N + 1 - a * nx),\ (N + 1 - a * ny) : (3N + 1 - a * ny)\right);$$

To update the phase change induced through mutation, the major change of implementation in the higher segmentation during each mutation operation should be:

$$New\_Efield\_H_x = Efield\_H_x + \left(\exp\left(1i * \frac{\pi}{2^n}\right) - 1\right) * (-1)^{Genome_H(MutationX1, MutationY1)} *$$

$$Efield_{2x}\left((M + 1 - a * nx) : (3M + 1 - a * nx),\ (M + 1 - a * ny) : (3M + 1 - a * ny)\right);$$

$$New\_Efield\_H_y = Efield\_H_y + \left(\exp\left(1i * \frac{\pi}{2^n}\right) - 1\right) * (-1)^{Genome_H(MutationX1, MutationY1)} *$$

$$Efield_{2y}\left((M + 1 - a * nx) : (3M + 1 - a * nx),\ (M + 1 - a * ny) : (3M + 1 - a * ny)\right);$$

$$New\_Efield\_H_z = Efield\_H_z + \left(\exp\left(1i * \frac{\pi}{2^n}\right) - 1\right) * (-1)^{Genome_H(MutationX1, MutationY1)} *$$

$$Efield_{2z}\left((M + 1 - a * nx) : (3M + 1 - a * nx),\ (M + 1 - a * ny) : (3M + 1 - a * ny)\right);$$

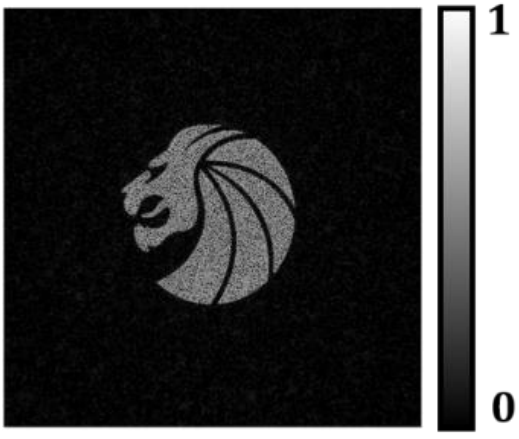

Figure S3. Simulated Binary Phase-controlled Meta-hologram based on SHEA. Emblem used with

permission from Seven Lions Music LLC.

As a typical proof-of-concept examples, Figure S4 shows the reconstructed simulated meta-hologram image of a 302×302 pixelated binary phase-controlled meta-hologram designed based on SHEA. The reconstructed binary phase-controlled meta-hologram is designed at $\lambda = 633nm$ wavelength's incidence, focal plane is designed at $z = 45\mu m$, while the dimension of each pixel is $300 \times 300nm$.

Part 6. Discussion on Absence of Comparison between GA and SHEA

According to previous reports,[1-2] where the authors applied genetic algorithm (they named it evolutionary algorithm, but the gist and process is the same as genetic algorithm) to solve inverse design problems for a 33×33 pixelated metasurface, they used 600 initial population. And their problem has $2^{33\times33} \approx 1.72 \times 10^{320}$ total possible solutions. Our problem has $2^{302\times302} \approx 1.72 \times 10^{26800}$ total possible solutions. Therefore, we infer that the appropriate initial population in our problem would be far more than 600 and would easily exceed the RAM of a regular personal computer. On the other hand, before launching the evolutionary process, as mentioned in the first part of the Supporting Information, we first need to calculate the complex electric fields from every nanopillar of the metasurface to every pixel of the target plane, which contains x, y and z components. If we directly save the complex electric fields in a matrix, the size of the matrix can be estimated as: $3 \times (302 \times 302 \times 605 \times 605) \times 8/(1024)^3 \approx 746.2$GB, which would far exceed the RAM of a personal computer. In order to solve this problem, we took advantage of the symmetric property of the light field penetrating through the Si nanopillar according to the Huygens-Fresnel principle, and only the electrical fields of the center nanopillar on the metasurface to the target plane need to be calculated and saved in a matrix. Therefore, the complex electrical fields coming from all the other nanopillars can be obtained from the matrix through translation operation by comparing the relative position of the nanopillar with the center one. And the total electric fields on the target plane can be obtained through adding up all the fields contributed from each nanopillar. Therefore, the size of the initial matrix can be reduced to $3 \times \big((2 \times 605 - 1) \times (2 \times 605 - 1)\big) \times 8/(1024)^3 \approx 33.5$MB. And we have proved that in SHEA and modified GA, when the initial population is only one, this strategy is very efficient. However, in GA, since the initial population is much larger than one, nested for loops (instead of matrix dot production) have to be implemented to calculate the reconstructed intensity profile and fitness value for every genome, which could be very time-consuming. In order to have a valid comparison between SHEA and GA, we tried to implement GA to design the same broadband Si meta-hologram shown in the main text based on a 300×300 pixelated metasurface with 100 initial population. However, it would take more than 11 hours to finish one generation on average on our computer, and convergence rate of GA is much slower even than that of modified GA. To further prove this, we make a comparison between GA and modified GA solving 33×33 pixelated meta-holograms so that the RAM won't be a limitation which affect the optimization performance. The dimensions of each pixel is $300\text{nm} \times 300\text{nm}$, and the target plane is at $z = 6\mu\text{m}$. The results are provided in figure S7.

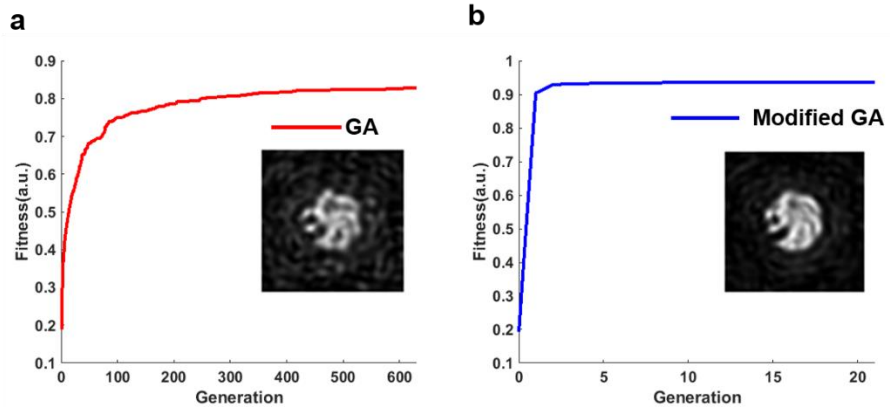

Figure S4. Comparison of GA and modified GA. (a) Convergence curve of GA. The inset picture is the reconstructed hologram image based on final optimized genome. (b) Convergence curve of Modified GA.

The inset picture is the reconstructed hologram picture based on final optimized genome. Emblem used with permission from Seven Lions Music LLC.

Figure S4 shows the comparison of GA and modified GA using correlation coefficient as the fitness function. In GA, the optimization process runs 600 generations and then reach a steady value, the simulation stops when the fitness value keeps unchanged for 30 generations. While in modified GA, the optimization process only needs 16 generations to reach a steady value (here, since the pixel number is small, we considered accuracy in 4 decimal places of the value of the fitness function for better optimized performance), and the simulation stops when the fitness value keeps unchanged for 3 generations. In GA, the average running time of one generation is around 61 seconds, while in modified GA, it only takes around 0.5 seconds to finish one generation. The final fitness value in GA is around 0.84, while the final fitness value in modified GA is around 0.93. Therefore, it can be concluded that when solving small-pixelated complex inverse problems, the convergence rate and computing speed of modified GA is better than that of GA. Thus we can infer that when dealing with complex large-pixelated inverse problems, SHEA should behave better than GA at convergence rate and computing speed, meanwhile, it has less stringent requirement of CPU speed and memory.

References:

1. Huntington, M. D.; Lauhon, L. J.; Odom, T. W., Subwavelength Lattice Optics by Evolutionary Design. *Nano Lett.* **2014,** *14*, 7195-7200.

2. Hu, J.; Liu, C.-H.; Ren, X.; Lauhon, L. J.; Odom, T. W., Plasmonic Lattice Lenses for Multiwavelength Achromatic Focusing. *ACS Nano* **2016,** *10*, 10275-10282.