

## Computer Generated Holograms of 3D Points Cloud

Vladimir CVILJUŠAC, Alan DIVJAK, Damir MODRIĆ

**Abstract:** Computer generated holography is a method of using numerical methods to simulate the physical processes underlying a real hologram's optical recording and reconstruction i.e. the method of digitally generating holographic interference patterns. As such, it represents substantial progress compared to classical holography. A major disadvantage of this approach is the absence of specialized software tools for computer generated holograms and extremely high computer resource consumption, resulting in an excessive computing time. The goal of this paper is to create a physically plausible computer model for generating holograms, with a large potential for calculation optimization and acceleration, as well as the integration of this model into simple hologram creation software. We made software that simulates wave optics using previously known physical model to generate a hologram with the possibility of connection with commercial 3D tools via .obj documents. Considering that there is a problem of high computing resources consumption and time we have incorporated in our software the possibility of a parallel calculation for the purpose of testing, further software development and customization to optimize the computer generation of holograms.

**Keywords:** computer generated holograms; digital content creation; point hologram; 3D point cloud; visualization

### 1 INTRODUCTION

Holography is the technique of recording and reconstructing light wave fronts without any prescribed amplitude and phase distribution, and therefore has found many applications. The production of such holograms has been discussed by Lee [1], Yaroslavskii and Merzlyakov [2] and Dallas [3]. Computational holography begins with a 3D numerical description of the object or scene to be imaged. Traditional, conventional holographic computation imitated the interference of optical holographic recording.

The physical principles of the hologram generation are well known [4, 5]. When holography was being developed in the 1960s, interference patterns were generated physically, using laser light and high-resolution photographic film. However, with the contemporary development of inexpensive, powerful computers, it has become increasingly feasible to generate holograms computationally.

Classical holography suffers from disadvantages limiting its usefulness [6]. In order to obtain high-quality classical hologram strict laboratory conditions need to be met, considering that the presence of small vibrations, impurities in the air and changes in temperature can seriously impair the hologram quality [7]. Most of these problems are eliminated with sophisticated and expensive equipment, wherein the size and content of hologram is limited. These disadvantages create the necessity for a different approach to hologram generation. Computer generated holography solves mentioned problems of the classic holography and enables new possibilities which were previously impossible.

Computational approach makes holography more popular by making it available to a wider range of users (any 3D graphic designer can generate holograms) and creates new opportunities for its application, as well as:

- Holograms are widely used for the information protection [8], the protection of graphic products against counterfeiting (securities, banknotes, identification documents, packaging, valuables...)
- Visual communications and multimedia presentations [9]: 3D models and scene visualization, 3D image and

video in medicine, architecture, mechanical engineering and design.

One stumbling block in computer generated holography is information overload. Since the optical holograms must contain details on the order of the wavelength of light, there is a huge amount of information to be accounted for. The computation engine for most hologram programs is designed to run on a standard workstation, which implies limited memory and computation speed. Thus, the engine makes considerable compromises in order to reduce computational demand. Simulation of physical principles governing hologram generation consumes a tremendously large number of numerical calculations. Each of the 3D model points interferes with every hologram pixel and very high pixel density is needed. The hologram 50×70 cm in size with the pixel density of 10,000 pixels per millimeter (ppmm) or 254,000 points per inch (dpi) has a total resolution of  $35 \times 10^{12}$  pixels. In comparison common print resolution on e.g. an office printer of 300 dpi for the same size is  $49 \times 10^6$  pixels, or about 700,000 times smaller. Furthermore, complex 3D scenes [10] may contain millions of points. For example, for a 3D scene consisting 100 million points it is necessary to calculate interference pattern of each model point with each hologram pixel, resulting in  $3.5 \times 10^{21}$  interference calculations.

The next disadvantage is the lack of the commercial software tools for hologram generation and reconstruction. There are, however, some computer-aided engineering tools that can perform wave optics simulation, but there is no specialized software for a holography, much less one that is accessible to wider range of users.

Computer generated holography is an area where broad and complex entangled physics and computer science is concerned, but from the point of view of the user there is no reason why the creation of computer generated holograms cannot be a simple matter. In the ideal case the user should simply load his 3D models into a hologram creation software, define the visual features of the model, determine the orientation and position of the model, reference beam and hologram in space and choose the type and hologram positions. After the calculation has been

performed, an image file of hologram is created which may be further used according to the user's necessities.

The aim of this paper was to create a physically plausible computer model for hologram generation, with a large potential for calculation optimization and acceleration, as well as the integration of this model into simple hologram creation software. The result is a software tool offering simpler and faster generation of holograms through a user-friendly graphic interface.

## 2 THEORY

Hologram computation is based on previously known method of elementary hologram generation (point hologram) [11]. In order to accelerate calculation we implement certain modifications. The reason for the selection and application of this calculation method is the fact that the point cloud models are exceptionally compatible with the elementary hologram computation.

It would make sense to calculate the diffraction of light starting from Maxwell's equations of the electromagnetic field. However, the coupling between the electric and magnetic vectors rapidly complicates the equation, and analytic solutions can only be found for highly symmetrical cases.

The issue can be greatly simplified by replacing the electric vector-wave equation  $E$  with a scalar electric field, while the magnetic component of the field is neglected. It is assumed that the diffraction does not affect the polarization of the incident wave.

These two assumptions are made as a basis for the simplified Kirchhoff diffraction integral known as the Fresnel diffraction integral (Fig. 1):

$$E(x, y, z) = \frac{e^{ikz}}{i\lambda z} \int_{\text{aperture}} E(x_0, y_0, z_0) \cdot e^{\frac{ik}{2z}[(x_z - x_0)^2 + (y_z - y_0)^2]} ds \quad (1)$$

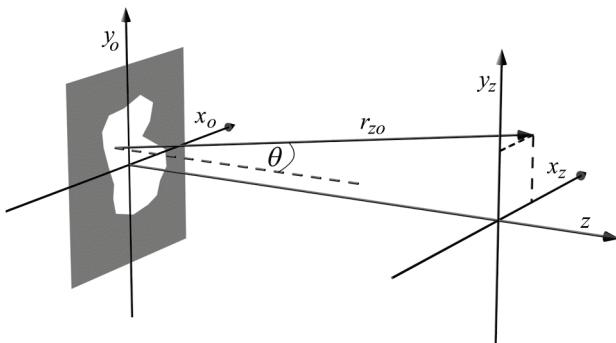


Figure 1 Illustration of Kirchhoff diffraction integral

The open portion of the aperture acts as a homogeneous source of the field  $E_o(x_o, y_o, z = 0)$ , as stated by Huygens' principle, and the field is zero in the opaque portion of the aperture.

$$E(x_z, y_z, z) = \sum_{\text{aperture}} [\text{incident field at } x_o, y_o, z = 0] \cdot [\text{wave propagation to } z : r_{zo}] \quad (2)$$

where is:

$$r_{zo} = \sqrt{z^2 + (x_z - x_o)^2 + (y_z - y_o)^2} \quad (3)$$

This equation is useful when  $z$  is large, compared to the wavelength.

### 2.1 The Hologram Calculation Method

Basic stages in the synthesis of computer generated holograms are: (i) formulating mathematical models of the object and of the usage of the hologram; (ii) computing the *mathematical hologram*, or distribution of the wave front amplitude and phase in the hologram plane; (iii) encoding samples of the mathematical hologram for recording them on the physical medium and (iv) fabrication of the hologram.

Computing the mathematical hologram simulates numerically wave propagation for the selected models of the object and of the hologram usage. Mathematical holograms are arrays of complex numbers that represent amplitudes and phases of hologram samples. For fabrication of holograms these numbers should be converted into an array of numbers that control optical properties of the physical recording medium used for recording the hologram.

The main principle of the hologram creation is the interference representing the superposition of two or more light waves of the same wavelength [12]. Depending on the phase difference between the waves, the interference may be constructive or destructive [13]. Constructive wave interference results in the local addition of waves and is shown with light (exposed) hologram areas while the destructive interference results with local subtraction and dark (unexposed) hologram areas. The created structure represents the encoded wave front of the holographically recorded object [14]. When the hologram is illuminated with an adequate light source (reference beam) the object wave front is reconstructed through diffraction.

When recording a hologram, light reflected from the object (object beam) loses collimated property of the laser beam with which it is illuminated, dispersing in all directions. The result of this is that every point of the object affects the brightness and interference pattern of each pixel of the hologram. This mutual influence results in an exceptionally large number of light interactions between the object and the hologram and represents the main problem of computer generated holography.

There are numerous methods of computer generated holography, each of them with their own benefits and disadvantages. The basis of this paper is a method of calculating the elementary hologram (point hologram), which is based on Huygens' principle, which states that every point of wave front is a source of a new spherical wave. This fact enables the treatment of the object surface as a cluster of coherent point light sources. Since point clouds are chosen as input data it is obvious that the elementary hologram method is the ideal method to use.

For each point of the model one elementary hologram is generated, that is, the reference beam interference (user defines the type and shape of the reference wave) and the spherical object wave of the model point. This process is

repeated for every model point until the hologram of the whole object has been created.

## 2.2 Calculation Principle

Hologram generation requires the following input parameters: position of the model points, position and dimension of the hologram, hologram pixel density, the reference beam shape and the wavelength of laser used for object illumination.

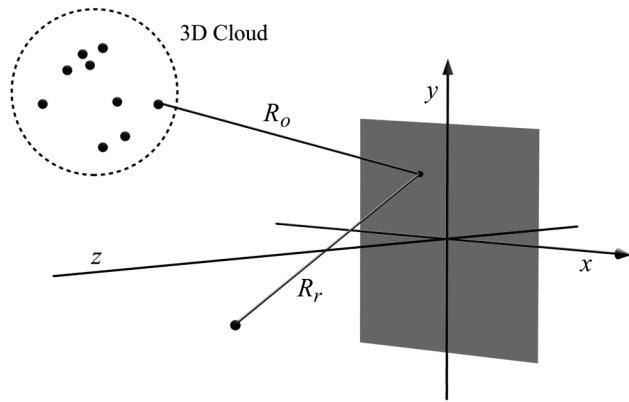


Figure 2 Hologram principle

Hologram consists of pixels and before starting calculation it is necessary to generate a matrix that contains positions of pixels in space. Hologram is defined by its width, height, pixel density and position in space. In this paper hologram is always positioned at the origin of coordinate system and it lies in  $XY$  plane, while the object is always positioned on the positive side of  $Z$  axis (Fig. 2).

The reference beam is defined by shape and radiation power that is defined by the user, and, according to some initial assumptions, is taken as a plane or spherical wave. The object is represented by a cluster of coherent point light sources where each point represents a separate spherical object wave. Each point has a position defined through  $X$ ,  $Y$  and  $Z$  coordinates which are created after loading 3D model into the program.

The interference calculation is based on the superposition of waves of the same wavelength, meaning that it is necessary to calculate maximum amplitude of a light wave at point  $Q(x, y, z)$  on hologram to equal

$$E(Q, t) = E_{\max}(Q) e^{i[\varphi(Q) - \omega t]} \quad (4)$$

The interference of two coherent beams is

$$E(Q, t) = E_{\max, 1}(Q) e^{i[\varphi_1(Q) - \omega t]} + E_{\max, 2}(Q) e^{i[\varphi_2(Q) - \omega t]} \quad (5)$$

and intensity equals the averaged amplitude. The electric field amplitude at a certain distance  $R$  from the light source depending on the phase  $\varphi$ :

$$\varphi_i(x, y, z) = \frac{2\pi}{\lambda} \left[ \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \right] \quad (6)$$

$$I_i = \left( \frac{E_{\max}}{R_i} \right)^2 ; i = o, r \quad (7)$$

Interference pattern of two coherent beams in a point  $Q(x, y, z)$  on hologram is:

$$I(Q) = \int E(Q, t) E^*(Q, t) dt = \\ = I_r + I_o + 2\sqrt{I_r I_o} \cos[\varphi_r(Q) - \varphi_o(Q)] \quad (8)$$

$$[\varphi_r(Q) - \varphi_o(Q)] = \\ = \frac{2\pi}{\lambda} \left[ \sqrt{\left( x - \frac{\Delta x}{2} \right)^2 + \left( y - \frac{\Delta y}{2} \right)^2 + \left( z - \frac{\Delta z}{2} \right)^2} - \right. \\ \left. - \sqrt{\left( x + \frac{\Delta x}{2} \right)^2 + \left( y + \frac{\Delta y}{2} \right)^2 + \left( z + \frac{\Delta z}{2} \right)^2} \right] \quad (9)$$

$$\Delta x = x_r - x_o, \Delta y = y_r - y_o, \Delta z = z_r - z_o$$

In order to obtain the resultant intensity at the position of a single pixel on the hologram, it is necessary to sum up the contributions of all pixels in the cloud that reach the respective pixel. For example, some of the points in the cloud do not have an optical connection (they are screened with points that form a side that is oriented towards the plane of the hologram) with particular pixels, so their contribution cannot be included in the final result.

$$I_k = \sum_{j=1}^n (I_k)_j \quad (10)$$

$k$  denotes a  $k^{\text{th}}$  pixel on the hologram, and  $n$  is the number of points in the cloud which "see" the  $k^{\text{th}}$  pixel, and depending on the geometry of the object  $n < m$  ( $m$  is the total number of dots in the cloud).

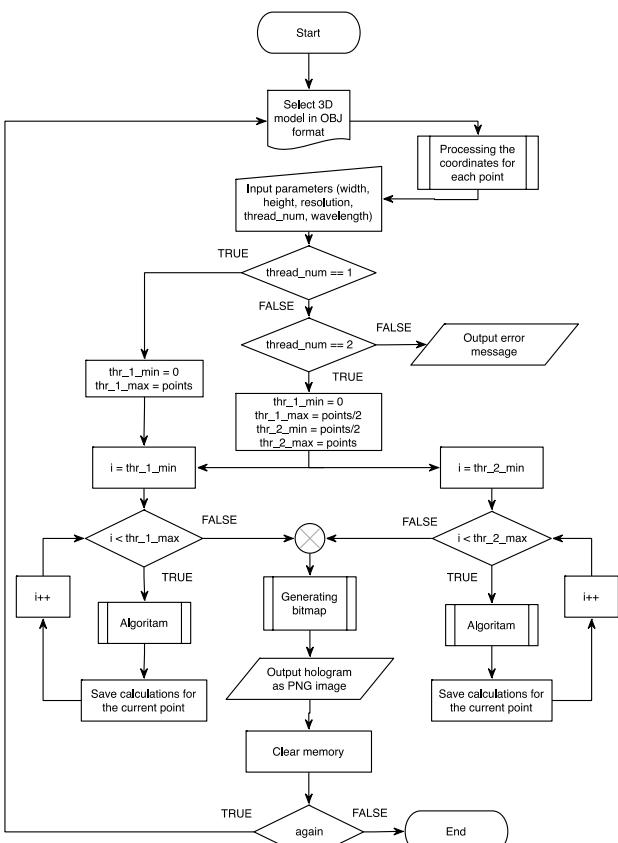
Production of the hologram can be simplified considerably if its transmittance has only two levels – either zero or one. After calculating the hologram the final step is tone mapping of intensities to image file. Binary holograms are generated so that white pixels represent the areas of the constructive interference and black pixels the areas of the destructive interference. For this it was necessary to define the threshold intensity. For the research purposes threshold  $t$  is defined as follows:

$$t = \frac{I_{\max} + I_{\min}}{2} \quad (11)$$

Intensity at the pixel, which is below this threshold is interpreted as binary 0, whereas the intensity above the threshold is interpreted as binary 1.

## 2.3 Computer Process Analysis

Holo 1.0.0 software is created in ActionScript 3.0 OOP (object-oriented programming) programming language using Starling and Feathers framework that enable the use of graphic processor acceleration. It is executed in the AIR (Adobe Integrated Runtime) runtime enabling us a wider range of target platforms, an important matter for the experimental part.

**Figure 3** Software flow diagram

Holo 1.0.0 software enables parallel calculation with two threads in order to execute calculation speed comparative tests and form preconditions for further development (Fig. 3). The user may choose manually whether to use one or two parallel threads as shown in the software flow diagram.

#### 2.4 Program Procedure

This version of software supports loading 3D models in .obj format. The .obj document is a textual type document containing the coordinates of the model points in the following format where  $v$  denotes vertex and following numbers are coordinates:

$v\ 1.0,\ 2.3,\ 4.5$ (e.g.  $v\ x,\ y,\ z$ )  
 $v\ 1.0,\ 2.3,\ 4.5$   
 $v\ 1.0,\ 2.3,\ 4.5$

While loading the 3D model, Holo 1.0.0 software uses the function to separate text documents into basic parts, extracts  $X$ ,  $Y$  and  $Z$  coordinates of each point of the model and stores them in the memory field. This sets the basic assumptions of the calculation.

##### 2.4.1 Division of Tasks to Threads

The hologram calculation requires that for every point of the model all the pixels of the hologram surface are calculated, and the division of the computation task to threads is done by assigning an equal number of model points to each thread.

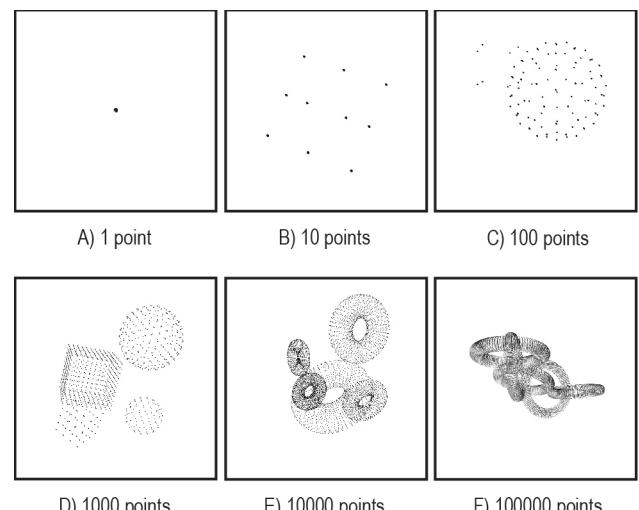
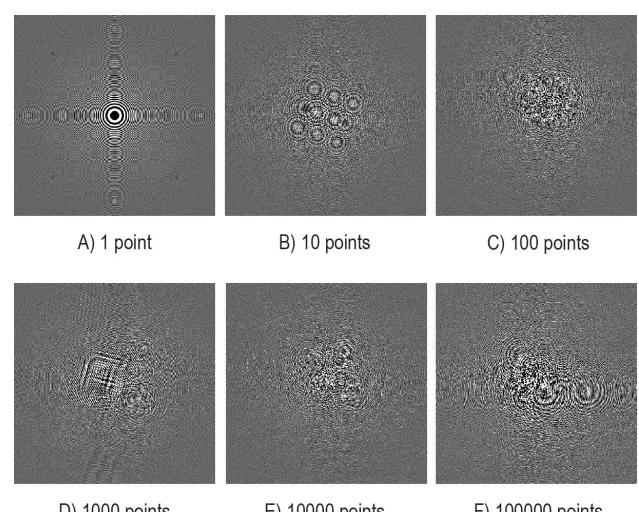
The calculation procedure with two threads is asynchronous meaning that each thread has different calculation time on which the user has absolutely no influence. The results of both parallel processes are at the end added giving the final result.

#### 2.4.2 Storing the Holograms

After the calculation has been completed, the result obtained is presented in a 2D matrix while intensity values for each pixel are transformed to 8-bit grayscale or binary bitmap (per user choice). The created bitmap is coded to PNG or JPG format depending on the user's preference and is then stored in the desired location on the computer.

### 3 EXPERIMENT

In order to establish different aspects of quality and usability of described hologram creation method, we generate several different holograms that differ by the number of points and the geometry of the cloud itself.

**Figure 4** Presentation of the used 3D models**Figure 5** Presentation of the computer generated holograms in experiment

The speed of hologram calculation is especially important, but examining the structure of the generated hologram is equally important as it gives an insight in interference pattern behavior under variation of input

parameters. The data on the appearance of the interference pattern can be used for further process of calculation optimization and acceleration, as well as automatic determination of optimal input parameters that simplifies software use for the user.

For testing purposes six 3D models [15, 16] were made with different complexity levels, which is reflected in the different number of points (1, 10, 100, 1000, 10000 and 100000 points) (Fig. 4). The models were created in LightWave 3D 2015 software for 3D modeling and stored in the .obj format. All holograms were calculated as if they were generated with a laser wavelength of 650 nm (Fig. 5).

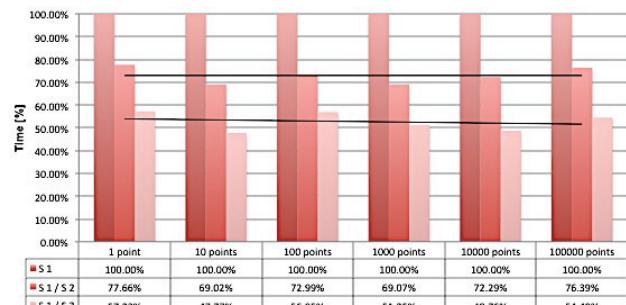
### 3.1 The Speed of the Hologram Calculation Depending on the Model Complexity

Calculation time is strongly correlated with the number of points in 3D model and number of pixels in the hologram. All 3D models were tested on three different computer systems (Tab. 1) and each had different execution time (Fig. 6). The hologram dimensions are 0.4×0.4 mm with pixel density of 5000 ppmm (pixel per millimeter).

**Table 1** Specification of the systems used in the calculation

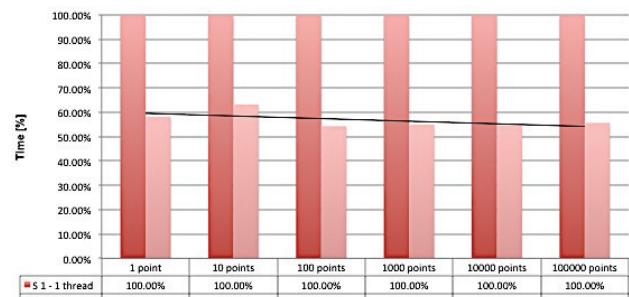
Label	S 1	S 2	S 3
Processor	Intel Core i7, 2,9 GHz	Core i7 260QM, 2,9 GHz	AMD PhenomX6 1100T, 3,7 GHz
RAM	8 GB 1600 MHz DDR3	8 GB 1600 MHz DDR3	16 GB 1600 MHz DDR3

The results show the expected linear increase of calculation time with the complexity of the model. Due to the very wide range of the obtained values, we used a logarithmic scale in the graph. The graph linearity (Fig. 8) on the logarithmic scale indicates an exponential increase in calculation time. The obtained results are very similar on all three computers.

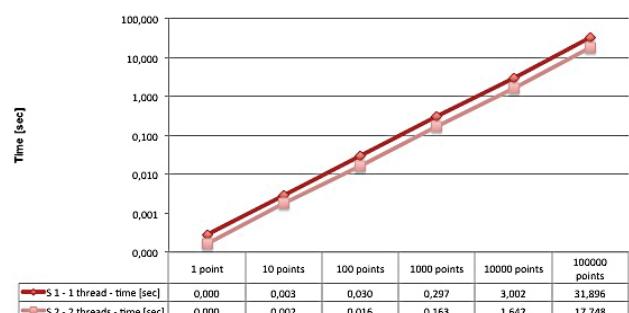


**Figure 6** Presentation of the comparison concerning the necessary time for the calculation of different models through three different systems

The use of two threads for calculation achieves the expected double acceleration and the obtained results are in accordance with the expected appearance of the interference pattern, proving that the physical model and its program code implementation give correct results. The graphs (Fig. 7 and Fig. 8) show the mean values vary very little (nearly horizontal black line) and it is clearly visible that equal acceleration is achieved. The average acceleration value on two threads is 56,8%, what we regard as substantial acceleration.

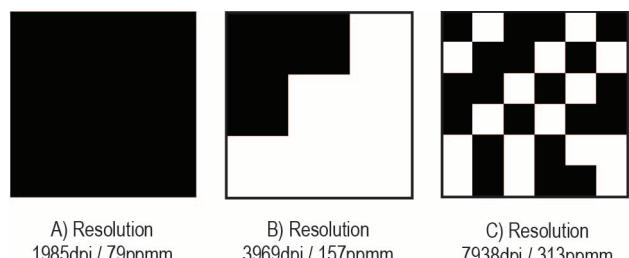


**Figure 7** Presentation of the comparison concerning the necessary time for the calculation on one and two parallel threads



**Figure 8** Presentation of the increase of calculation time in relation to model complexity

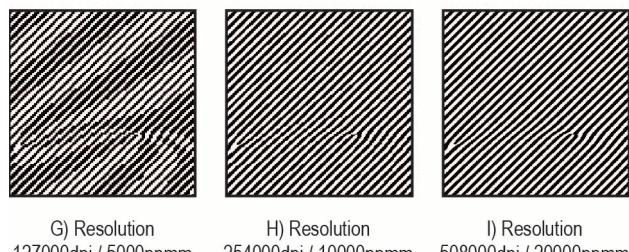
The speed of the calculation as a function of model complexity increase was also measured and investigated.



A) Resolution 1985dpi / 79ppmm  
B) Resolution 3969dpi / 157ppmm  
C) Resolution 7938dpi / 313ppmm



D) Resolution 15875dpi / 625ppmm  
E) Resolution 31750dpi / 1250ppmm  
F) Resolution 63500dpi / 2500ppmm



G) Resolution 127000dpi / 5000ppmm  
H) Resolution 254000dpi / 10000ppmm  
I) Resolution 508000dpi / 20000ppmm

**Figure 9** Interference pattern at different pixel density

### 3.2 Optimal Hologram Pixel Density

Exceptionally high hologram pixel density is the main reason for a large consummation of the computational resources. Defining optimal pixel density is most important

for economizing calculation time. 3D model with 100,000 points is used as it generates the most complex interference pattern. The hologram width and height are constant while the pixel density changes from 79 to 20,000 ppmm (Fig. 9). The optimal pixel density is achieved when the difference between the two holograms is small enough that it does not cause deterioration of the visual quality of interference pattern shape.

The results (Fig. 9) clearly demonstrate the greatest problem concerning computer generated holography. At pixel density of around 2000 dpi, normally reserved only for the highest quality classic image printing systems, the generated hologram does not contain any usable interference structures. Similar situation continues till the density of 127,000 dpi is reached. It can be concluded that the optimal resolution is at approximately 15,000 ppmm or 380,000 dpi. This pixel density generates 225 million pixels on every square millimeter of hologram. In comparison, Canon 5DSR, one of the highest resolution cameras on the market today, creates images with just 50 million pixels.

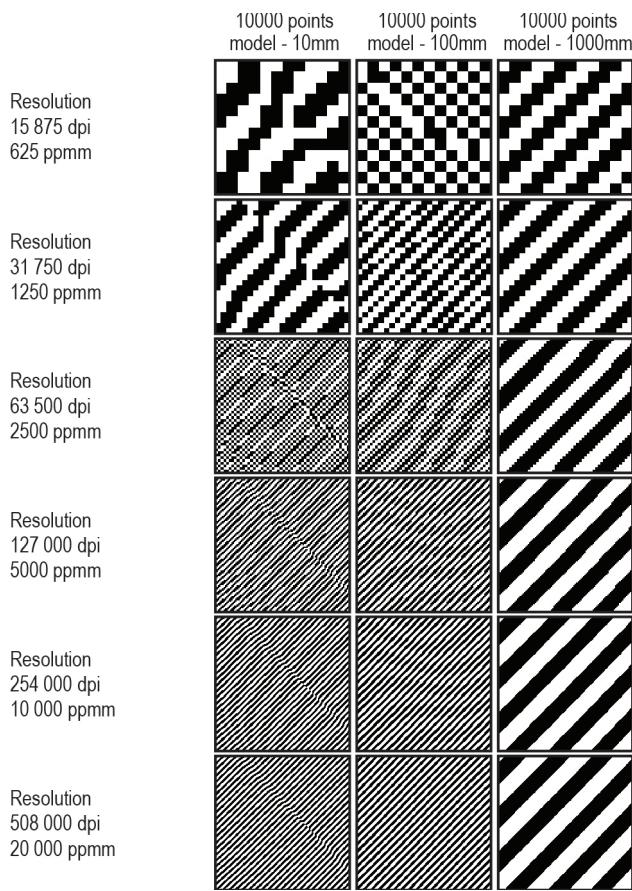


Figure 10 Interference pattern at different distance object from hologram

### 3.3 The Influence of the Distance of the 3D Models on the Optimal Pixel Density

The user determines positioning of 3D model in space. It is expected that increasing the distance of the model to hologram will result in reduced optimal pixel density because of slower intensity change in interference patterns between reference and object wave at the same wavelength. Most complex 3D model (100,000 points) is used during testing and three different model distances

from hologram are tested (10, 100 and 1000 millimeters). These values were chosen because they represent real situations when creating a hologram. The remaining parameters of hologram creation remain unchanged.

The results (Fig. 10) show clearly that the increase of distance influences the optimal pixel density. At the minimal distance between the model and the hologram the optimal density is 15,000 ppmm while at the maximum distance the required density is only 1,250 ppmm.

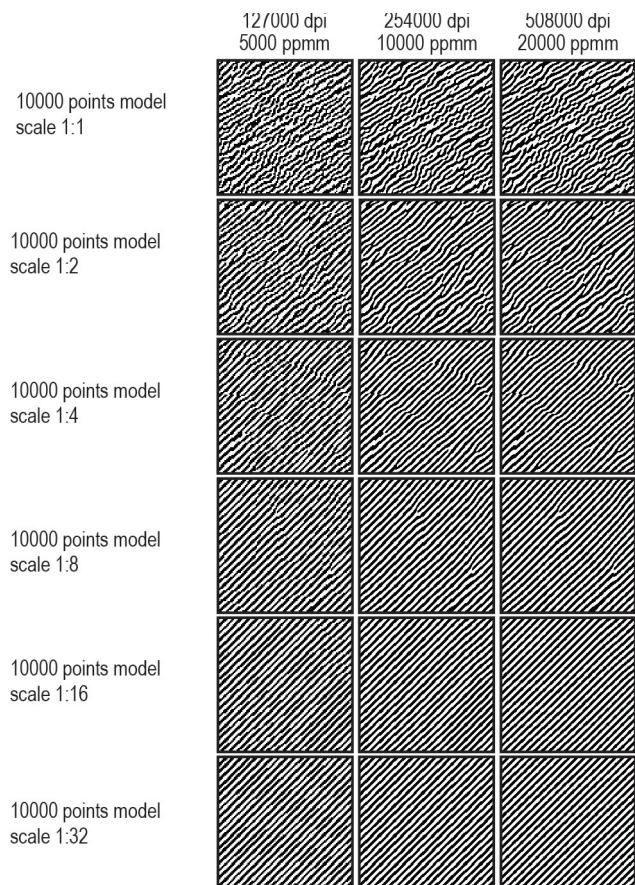


Figure 11 Interference pattern at different object size in regard to hologram

### 3.4 Influence of Object Size in Regard to Hologram on the Optimal Pixel Density

Decrease of model size brings the reflected wave front nearer to a point source. In such case interference pattern becomes denser from the hologram's middle area towards its edge. This also means that it is necessary to generate a hologram with a greater pixel density. It is expected that increase in object size would result in less dense interference pattern that would enable hologram generation at lower pixel density, which accelerates calculation and saves time. In order to research the dependence of the object size in regard to pixel density, most complex model consisting of 100,000 points is used. The size of object is reduced in regard to hologram, while other parameters remain constant.

The results (Fig. 11) show that the distance between the interference pattern does not change, what is in contrast to the expected results. However, the shape of the pattern does change; pattern of larger objects has a more complex shape and is well defined even at lower pixel density, while the pattern of smaller object is composed of almost parallel

lines whose quality quickly degrades at lower pixel density. When the model size is equal to the size of the hologram the optimal pixel density is around 7,000 ppmm, while the models with dimensions smaller than 1/32 of the hologram have an optimal pixel density of 15,000 ppmm.

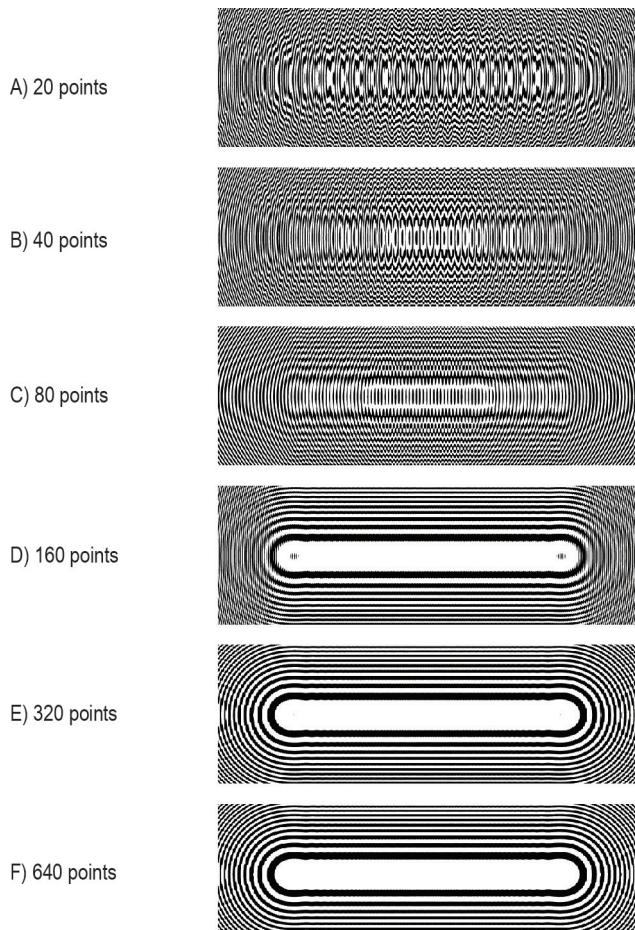


Figure 12 Interference pattern at different point density of object

### 3.5 Optimal 3D Model Point Density for Continuous Surface Appearance

The described model of hologram generation uses point clouds for the 3D model representation. This approach has many advantages such as simple and fast calculation, but it is not able to present continuous surfaces like polygonal models. The intention is to adapt the model of the hologram calculation so it can create continuous surfaces, but first it is necessary to determine the required point density in 3D space in order to get the impression of a continuous surface. The optimal point density threshold is determined by generating holograms of a line model (Fig. 12).

The number of points that define the line is gradually increased for each hologram. With the increase of the number of points the spherical wave fronts are merged together in one single wave front. On the hologram this is seen as a cluster of elementary holograms joined together in a continuous structure. Optimal point density is reached when the increase of the number of points results with a continuous interference pattern.

Results show that in the given case the sufficient density of 3200 points per millimeter is enough to achieve a continuous line. In case of polygonal models it would be

necessary to have approximately 10 million points per square millimeter to achieve the same result. By moving the object away from the hologram, the distance between the interference patterns is increased and it can be concluded that the elementary holograms would start to merge into one single wave front sooner. This will be a matter for a further research. This point is exceptionally important in regard to the optimization of hologram generation and the mutual dependence between the object distance and the point density will be further investigated.

## 4 CONCLUSION

The computer generated holograms represent a significant and interesting area of physics whose application has a potential to drastically change the way the information is presented and transferred. With the daily increase in the available computing power and new algorithms we are approaching the moment when the generation of high-quality holograms of complex model begins to be possible and available to a wider range of users. The examples such as the holographic screens and projectors, holographic optical elements and holographic print are only a few of the numerous possibilities that will be realized in near future. The use of holography in the areas of medicine, architecture, multimedia, security features (e.g. banknotes, securities, etc.), consumer electronics, etc. is currently limited by the disadvantages of classic holography. Computer generated holography eliminates many problems of classical holography, however, at this time, fails to resolve remaining problems, such as printing and display of the hologram.

The results obtained show how the described physical model exactly simulates the spread of the electric field and the subsequent interference. The computer model implementation clearly shows good performance and the results suggest that this is high quality solution without potential problems such as stability or memory leaks across three different systems. Using two threads achieves double speedup, which further verifies physical model and program code quality.

The obtained calculation speed is relatively low considering the quantity of the calculation associated with creating holograms of useful dimensions and considering the point count of complex 3D objects, rendering the use of presented software impractical. However, further development of this software will serve as the base for computer generated holography platform and as a validation of physical and computer approach. It should also be noted that no optimizations are implemented at this moment. The results show that it is possible to realize numerous methods of optimizations, either through further development of the physical model or by performing the calculation on multicore processors or graphics processors, which will also be one of the topics of our future researches.

Testing of the optimal pixel density shows that it is possible to obtain a large calculation speedup through the 3D model structure analysis and adaptation of the calculation parameters according to the analysis results. The achievement of a continuous surface appearance is also exceptionally important for display fidelity as well as the calculation acceleration. Optimizations through the

model density reduction are also a very interesting topic for further research.

## 5 REFERENCES

- [1] Lee, W. H. (1989). Holographic optical head for compact disk applications. *Optical Engineering*, 28, 650-653. <https://doi.org/10.1117/12.7977015>
- [2] Yaroslavskii, L. P. & Merzlyakov, N. S. (1980). *Methods of Digital Holography*. New York: Consultants Bureau, Plenum Publishing Co. <https://doi.org/10.1007/978-1-4757-5827-6>
- [3] Dallas, W. J. (1980). Computer-generated holograms. *The Computer in Optical Research, Topics in Applied Physics*, vol. 41, ed. B. R. Frieden, Berlin: Springer-Verlag, 291-396.
- [4] Gabor, D. (1948). A New Microscopic Principle. *Nature*, 161, 777-778. <https://doi.org/10.1038/161777a0>
- [5] Gabor, D. (1949). Microscopy by Reconstructed Wavefronts. *Proceedings of the Royal Society*, 197, 454-487. <https://doi.org/10.1098/rspa.1949.0075>
- [6] Denisyuk, Y. N. (1962). On the Reflection of Optical Properties of an Object in a Wave Field of Light Scattered by it. *Doklady Akademii Nauk SSSR*, 144, 1275-1278.
- [7] Leith, E. N. & Upatnieks, J. (1962). Reconstructed Wavefronts and Communication Theory. *J. Opt. Soc. Am.*, 52, 1123-1130. <https://doi.org/10.1364/JOSA.52.001123>
- [8] Upatniek, J. & Leonard, C. (1969). Diffraction Efficiency of Bleached Photographically Recorded Interference Patterns. *Applied Optics*, 8, 85-89. <https://doi.org/10.1364/AO.8.000085>
- [9] Benton, S. A. (1977). White Light Transmission/Reflection Holography. *Applications of Holography and Optical Data Processing*, 401-409. <https://doi.org/10.1016/B978-0-08-021625-6.50046-4>
- [10] Lafarge, F. & Mallet, C. (2012). Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision*, 99(1), 69-85. <https://doi.org/10.1007/s11263-012-0517-8>
- [11] Blanche, P. A., Bablumian, A., Voorakaranam, R., Christenson, C., Lin, W., Gu, T., Flores, D., Wang, P., et al. (2010). Holographic three-dimensional telepresence using large-area photorefractive polymer. *Nature*, 468, 80-83. <https://doi.org/10.1038/nature09521>
- [12] Kozma, A. & Zelenka, J. S. (1970). Effect of film resolution and size in holography. *Journal of the Optical Society of America*, 60, 34-43. <https://doi.org/10.1364/JOSA.60.000034>
- [13] Koreshev, S. N., Gromov, A. D., & Nikanorov, O. V. (2012). Modernized Software Complex for Synthesis and Reconstruction of Fresnel Holograms-Projectors. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 82(6), 12-17. (in Russian)
- [14] Slinger, Ch., Cameron, C., Stanley, M. (2005). Computer-Generated Holography as a Generic Display Technology. *Computer (IEEE)*, 38(8), 46-53. <https://doi.org/10.1109/mc.2005.260>
- [15] Reeves, W. T. (1983). Particle Systems - A Technique for Modeling a Class of Fuzzy Objects. *Computer Graphics*, 17(3), 359-376. <https://doi.org/10.1145/964967.801167>
- [16] Rubin, S. M. & Whitted, T. (1980). A 3-Dimensional Representation for Fast Rendering of Complex Scenes. *Computer Graphics*, 14(3), 110-116. <https://doi.org/10.1145/965105.807479>

## Contact information:

### Vladimir CVILJUŠAC

University of Zagreb, Faculty of Graphics Arts,  
Getaldićeva 2, 10 000 Zagreb, Croatia  
E-mail: vladimir.cvijusac@grf.hr

### Alan DIVJAK

Intel d.o.o.  
Korčulanska 6, 10 000 Zagreb, Croatia  
E-mail: alan.divjak@gmail.hr

### Damir MODRIĆ

University of Zagreb, Faculty of Graphics Arts,  
Getaldićeva 2, 10 000 Zagreb, Croatia  
E-mail: damir.modric@grf.hr