

文章编号:1004-4213(2010)03-0529-4

一种菲涅尔全息图的快速算法*

李勇,许富洋,金洪震,王辉

(浙江师范大学 信息光学研究所,浙江 金华 321004)

摘要:通过分析菲涅尔全息图计算模型,发现大量的平方运算和三角函数运算是计算速度的瓶颈,提出了一种菲涅尔全息图的快速计算方法.根据菲涅尔全息图的数学模型,利用三角函数的和差化积公式,将全息图的数学模型变换为由仅与水平或垂直方向有关的独立分量,通过四则运算来表示.在菲涅尔全息图的计算中,用少量的三角函数、平方运算和大量的四则运算代替原来大量的三角函数和平方运算,减少了全息图的运算量.实验表明,采用该算法后全息图的计算速度提高了 9 倍以上.

关键词:信息光学;计算机全息;三维显示;全息显示;快速计算

中图分类号:O438.1

文献标识码:A

doi:10.3788/gzxb20103903.0529

0 引言

全息显示技术是最有前途的逼真三维显示技术之一.全息图的数字化工作一直是人们关注的议题. Schnars 等^[1]提出用 CCD 元件作为全息图的记录材料得到数字全息图,并用计算机获取物光波波前的方法.但该方法对记录系统要求高,而且仅仅是激光全息图的数字化,不可避免的相干噪音对于三维显示来说不够理想.而将全息过程数字化的计算机全息图是一种更为广义的数字全息图^[2],使得全息图制作更具灵活性,所以一直受到人们的特别关注^[3-4].目前全息三维显示技术正朝着全息影视方向发展^[5].数字全息三维显示技术实用化的一个关键问题是全息图的快速计算.为此,人们提出了一些快速算法提高全息图计算速度^[6-9].然而到目前为止,要计算一幅实用的全视差、大视角的数字全息图需要几十分钟到几十小时甚至更长时间.数字全息图的计算速度仍然是其实用化的一个瓶颈.本文根据菲涅尔全息图的数学模型,提出了一种菲涅尔全息图快速算法.它通过三角公式变换,将全息图的计算分解为水平和垂直方向计算结果的四则运算组合的形式,大大减少了计算量.实验表明,该算法是有效的.

1 原理

1.1 计算机全息图制作原理

计算机制作全息图通常采用如图 1 的模型,图中坐标系采用右手系,全息面 H 位于 xy 坐标平面上. $ABCD$ 为场景中的一个物体,它自身发射或反射光波到全息面上.因为反射物体可以等效为自发光物体,这里将场景看作是自发光的.由于数字计算机只能处理离散数据,制作计算机全息图首先要将场景按显示分辨率要求进行离散化,同时全息面上的点也要离散化.这样,场景中每一点发出的光也是角度离散化的“光线”.设离散化后场景中第 i 点 (x_i, y_i, z_i) 发出的“光线”在全息面上的复振幅分布为

$$O_i(x, y) = \frac{a_i}{r_i(x, y)} \exp \left\{ j \left[\frac{2\pi}{\lambda} r_i(x, y) + \varphi_i \right] \right\} \quad (1)$$

式中 $r_i(x, y) = \sqrt{(x-x_i)^2 + (y-y_i)^2 + z_i^2}$ 为物点到全息面上点 (x, y) 的距离, λ 为波长, $a_i/r_i(x, y)$ 是振幅, φ_i 为初相位,通常取随机值.为分析方便,

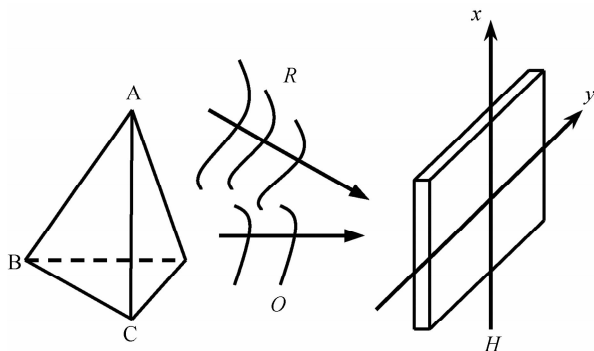


图 1 计算机全息图制作模型
Fig. 1 Computational model of CGH

*国家自然科学基金(60877002)和浙江省自然科学基金(Z1080030)资助

Tel:0579-82298859

Email:liyong@zjnu.cn

收稿日期:2009-07-08

修回日期:2009-09-24

这里采用平面波为参考光,结果同样适用于以球面波为参考光的情形.全息面上参考光复振幅分布可以表示为

$$R(x,y)=\text{bexp}\{j\Psi(x)\} \quad (2)$$

式中, $\Psi(x)=\frac{2\pi}{\lambda}x\sin\theta$, θ 为参考光入射角.在实际应用中,可以将物光的振幅分布近似看成常数.采用双极强度计算方法^[9]时,全息面上干涉光强分布可表示为

$$I(x,y)=A+\sum_{i=1}^N a_i \cos\left\{\frac{2\pi}{\lambda}r_i(x,y)+\varphi_i-\Psi(x)\right\} \quad (3)$$

式中 A 为直流偏置,使强度非负, N 为场景中的总物点数.

1.2 菲涅尔全息图分离变量原理

在菲涅尔近似下,式(3)第二项可以写成

$$I'(x,y)=\sum_{i=1}^N a_i \cos\left\{\frac{2\pi}{\lambda}\left[z_i+\frac{(x-x_i)^2+(y-y_i)^2}{2z_i}\right]+\varphi_i-\Psi(x)\right\} \quad (4)$$

从式(4)可以看出菲涅尔全息图计算中花费时间最多的是平方和余弦函数值的计算.对于尺寸为 M 点 \times N 点的数字全息图,计算每个物点的干涉光强分布需要 $4MN$ 次乘法运算(除法可以看成乘法),和 MN 次余弦运算.如果能对式(4)进行变换,减少余弦和乘法运算次数,就可以提高总运算速度.对式(4)进行变换,定义

$$\begin{aligned} I'_i(x,y) &= a_i \cos\left\{\frac{2\pi}{\lambda}\left[z_i+\frac{(x-x_i)^2+(y-y_i)^2}{2z_i}\right]+\varphi_i-\Psi(x)\right\} \\ &= a_i \cos\left\{\frac{2\pi}{\lambda}\left[z_i+\frac{(x-x_i)^2}{2z_i}\right]+\varphi_i-\Psi(x)+\frac{\pi}{\lambda}\cdot\frac{(y-y_i)^2}{z_i}\right\} \end{aligned} \quad (5)$$

令 $\varphi_i(x)=\frac{2\pi}{\lambda}\left[z_i+\frac{(x-x_i)^2}{2z_i}\right]+\varphi_i-\Psi(x)$, $\varphi_i(y)=\frac{\pi}{\lambda}\cdot\frac{(y-y_i)^2}{z_i}$, 则式(5)可以写成

$$I'_i(x,y)=a_i \cos\{\varphi_i(x)+\varphi_i(y)\} \quad (6)$$

根据三角函数和差化积公式,式(6)可以写成

$$\begin{aligned} I'_i(x,y) &= a_i \{\cos[\varphi_i(x)]\cdot\cos[\varphi_i(y)]- \\ &\quad \sin[\varphi_i(x)]\cdot\sin[\varphi_i(y)]\} = \\ &\quad a_i [I'_{ci}(x)\cdot I'_{ci}(y)-I'_{si}(x)\cdot I'_{si}(y)] \end{aligned} \quad (7)$$

式中 $I'_{ci}(x)=\cos\{\varphi_i(x)\}$, $I'_{ci}(y)=\cos\{\varphi_i(y)\}$, $I'_{si}(x)=\sin\{\varphi_i(x)\}$, $I'_{si}(y)=\sin\{\varphi_i(y)\}$. 由式(7)可以看出,全息图可以分解成若干个只与 x 或 y 有关的独立分量.分析全息图计算过程,可以发现其中有大量关于这些独立分量的冗余计算,去除这些冗余是提高速度的关键.实际上,这些分量只需要计算一行、一列.因此,在计算全息图时只在第一行和第

一列计算这些分量,即 $I'_{ci}(x)$ 、 $I'_{si}(x)$ 、 $I'_{ci}(y)$ 、 $I'_{si}(y)$, 并将结果保存在内存中.在计算全息图其它位置时,利用式(7)通过乘、加法运算将这些对应的预先保存的独立分量组合起来即可.这样,对于尺寸为 M 点 \times N 点的计算机全息图,只在计算第一行和第一列的独立分量时需要 $4\times(M+N)$ 次余弦(正弦运算可以看成余弦运算)运算,其它位置只有乘法和加法运算.计算式(7)需要 $4M+2N+2MN$ 次乘法,与计算式(4)相比,乘法运算量减少了 $2MN-(4M+2N)$ 次,余弦运算量减少了 $MN-4(M+N)$ 次.总计算速度得到了提高.

2 实验及结果分析

采用 C++ 语言编写了全息图计算程序,在 PC 机(CPU: Intel E4500@2.2GHz,内存:2.0GB)上进行实验,分别用式(4)和式(7)计算.首先研究了全息图尺寸固定,物点数变化时,两种算法的计算速度对比情况.全息图的尺寸为 1 200 点 \times 1 200 点,物点数分别为 1、2、50、400、800 点.得到实验数据如表 1.从表中可以看出,当物点数增加到一定程度以后,两种算法的速度比稳定在 10 倍左右.然后研究了物点数固定,全息图尺寸变化时两种算法的计算速度对比情况.物点数为 50 点,实验结果如表 2.从表中可以看出,当全息图尺寸增大到一定程度以后,两种算法的速度比稳定在 10 倍左右.

表 1 物点数变化时计算时间比较

Table 1 Comparison of computational time of different object points

Algorithm	Object points				
	1	2	50	400	800
Eq. (4)	0.082 4 s	0.156 3 s	3.685 s	29.490 s	58.799 s
Eq. (7)	0.015 6 s	0.022 5 s	0.358 5 s	2.844 5 s	5.616 5 s

表 2 不同全息图尺寸计算时间比较

Table 2 Comparison of computational time of different hologram size

Algorithm	Hologram size/(points)			
	120 \times 120	120 \times 1 200	1 200 \times 1 200	1 200 \times 12 000
Eq. (4)	0.037 1 s	0.367 1 s	3.685 s	36.765 s
Eq. (7)	0.004 49 s	0.035 8 s	0.358 5 s	3.585 7 s

最后,计算了一个通过三维扫描得到的真人头部数据的全息图.物点数为 90 000 点,全息图尺寸为 4 096 点 \times 1 024 点.计算时间为 31 分.图 2 为模型的屏幕显示截图、全息图片段和激光再现像照片.由于全息图再现像比较难拍摄,实际观察到的再现像比图中的清晰.

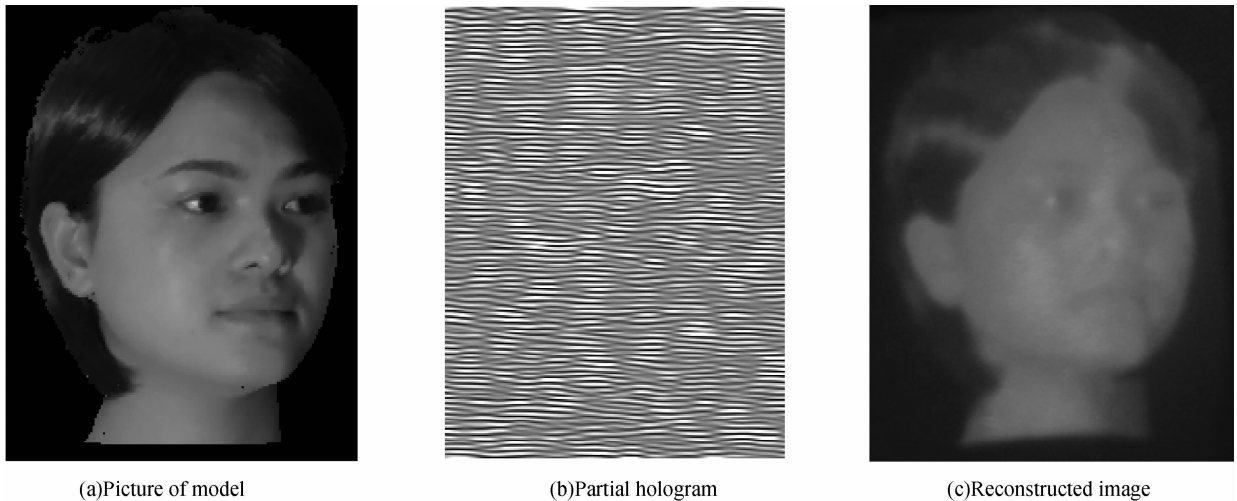


图2 实验结果

Fig. 2 Experimental results

3 结论

数字全息图的计算速度是其实用化的瓶颈之一.通过分析菲涅尔全息图数学模型,利用三角变换公式,将全息图分解成只与 x 或 y 方向有关的独立分量.从中发现全息图计算过程中的大量冗余运算.提出先计算一行、一列的独立分量,并保存在内存中.然后在其它位置的全息图计算时,取内存中与行号和列号对应的独立分量,将它们通过乘法、加法运算组合起来得到该点光强.这样去除了计算过程中的大量冗余运算,大大提高了计算机全息图的计算速度.实验表明,提出的算法比未变换前的算法只增加了少量内存需求量,而计算速度提高了9倍以上.

参考文献

- [1] SCHNARS U, KREIS T M, JÜPTNER W P O. Digital recording and numerical reconstruction of holograms; reduction of the spatial frequency spectrum[J]. *Opt Eng*, 1996, **35**(4): 977-982.
- [2] WANG H, SU X Y, LI Y, *et al.* Visualization using computer-generated hologram (CGH) based on three-dimensional shape measurement data of object[J]. *Acta Optica Sinica*, 2003, **23**(3): 284-288.
- [3] LESEBERG D. Computer generated holograms; cylindrical, conical, and helical waves[J]. *Appl Opt*, 1987, **26**(20): 4385-4390.
- [4] PASTOOR S, WÖPKING M. 3-D displays; a review of current technologies[J]. *Displays*, 1997, **17**(2): 100-110.
- [5] LUCENTE M, St HILAIRE P, BENTON S A, *et al.* New approaches to holographic video[C]. *SPIE*, 1993, **1732**: 377-386.
- [6] CAI Xiao-ou, WANG Hui, LI Yong. Study on the kinoform of frequency-change sampling[J]. *Acta Photonica Sinica*, 2005, **34**(1): 114-117.
- [7] JIN Hong-zhen, LI Yong, WANG Hui. Fast calculation of kinoform by use of difference method [J]. *Opto-Electronic Engineering*, 2001, **28**(4): 30-33.
- [8] LI Yong, JIN Hong-zhen, WANG Hui, *et al.* Research for horizontal parallax only kinoform in 3D display[J]. *Chinese Journal of Lasers B*, 2002, **B11**(6): 465-468.
- [9] LUCENTE M. Interactive computation of holograms using a look-up table[J]. *J Electron Imaging*, 1993, **2**(1): 28-34.

A Fast Computation Method for Fresnel Hologram

LI Yong, XU Fu-yang, JIN Hong-zhen, WANG Hui

(*Institute of Information Optics, Zhejiang Normal University, Jinhua, Zhejiang 321004, China*)

Abstract: The mathematical model of Fresnel hologram is analyzed, and it is found that the speed bottleneck of generating CGH is the operation of square and trigonometric function. A method for fast computation of Fresnel hologram is proposed. The mathematical model of Fresnel hologram is transformed to the form which is expressed as the four arithmetic operations of components related to horizontal direction and vertical direction of hologram, according to trigonometric identity. Then, the great amount of operation of square and trigonometric function is replaced by little amount of operation of square and trigonometric function and great amount of arithmetic operation. The computational amount of Fresnel hologram is reduced and the computational speed will be increased clearly. The experimental results show that the speed of Fresnel hologram computation is accelerated over 9 times by using the method of separation of variables.

Key words: Information optics; Computer-generated hologram; Three-dimensional display; Holographic display; Fast computation



LI Yong was born in 1972. He received his Ph. D. degree from Sichuan University in 2006. Now he works as an associate professor, and his research interests focus on optical information processing, optical 3D sensing and 3D display.