

全息图

光学全息

计算机

CGH

计算机科学2000Vol. 27No. 1

9-12

计算机生成全息图的研究与实现

Research of Computer-Generated Holography

杨 钦 徐永安 王元鹏

TB877

(北京航空航天大学计算机科学与工程系 北京100083)

Abstract A survey of computer-generated holography is presented in this paper. The methods of the fringe computation are analyzed. A computer-generated holography is realized by the diffraction-specific computation and the normal equipments.

Keywords Holography, Interference, Diffraction, Fringe

1. 引言

计算机生成全息图(Computer-Generated Holography, 简称 CGH)是继光学全息之后于1965年由 Kozma 和 Kelly 提出的^[1], CGH 是计算机技术与全息技术的结合, 为真三维显示技术开创了全新的局面。但由于受计算机运算速度、容量、高分辨率显示器以及传输带宽等因素的约束, 七十、八十年代 CGH 的研究未取得大的进展, 1989年, 美国麻省理工学院(MIT)媒体实验室以 S. A. Benton 为首的空间成像研究小组开始对 CGH 展开研究, 在美国 ARPA (Advanced Research Project Agency)、NSF (National Science Foundation) 和 IBM、NEC、Kodak、Hughes Aircraft、Intel、Philips、Sharp、Sony、Sun、Microsoft 等世界著名企业和组织的大力资助下, 在 CGH 方面取得了突破性的进展, 于九十年代初生成了真彩、实时的三维全息图^[2~4], 使 CGH 朝着实用化迈出了一大步。

CGH 的主要困难在于全息图的大数据量。要生成一幅100mm×100mm的全视差的全息图像, 其数据采样量大概是100G(假定使用氮氛激光, 其采样间距为波长的一半, 则整个数据量为 $(100/(6.328 \times 10^{-4}/2))^2 \approx 100 \times 10^6 \approx 100G$), 如此巨大的数据量使计算机生成全息特别依赖于硬件的 CPU 速度和高分辨率显示器, MIT 媒体实验室研究 CGH 时采用了昂贵的实验设备, 本文采用基于衍射的条纹生成法, 给出一个基于现有硬件条件的 CGH 实现方案。

2. 计算机生成全息图的原理

光学全息是根据干涉原理, 利用光学器件将物体发出的光波以干涉条纹记录成图, 这样的图称为“光学全息图”, 当用光波照射全息图时, 根据衍射, 能再现原

始物体的光波, 从而形成与原物体逼真的三维像。计算机生成全息图的核心是用程序替代光学器件生成全息条纹, 最直接的方法是模拟光学全息, 根据干涉原理在计算机中实现物光波和虚拟参考光的叠加, 这种基于干涉的条纹生成法, 能够达到光学全息的效果, 但存在光学全息的噪声, 计算速度慢, 不能编码等缺陷。为了克服这些缺陷, MIT 的 Mark Lucente 提出了基于衍射的条纹生成方法, 这种方法是受到全息条纹的作用的启发而提出的。全息条纹的作用是将光线朝某一个特定的方向衍射。因此, 要想衍射光线在透过全息平面上的一点后朝某个方向衍射, 只要在该点有某种形式的条纹即可, 基于衍射的条纹生成方法就是提供了一个反向计算全息条纹的方法。该方法事先计算出一组独立的、承担特定衍射任务的基本条纹, 当计算一个物体的全息条纹时, 按需从基本条纹中挑出一些符合要求的进行叠加。

CGH 对硬件设备的要求很高, 在一般硬件条件下必须通过减少数据量来进行 CGH 研究。减少数据量的方法有两种:

(1) 消除竖直视差。人眼绝大部分时间是在水平方向上观察物体, 对竖直方向的视差不敏感。这样就可以在竖直方向上采用较低的分辨率2lp/mm; 而在水平方向上采用1000lp/mm的高分辨率。经过这样的简化, 一幅全息图就成为一系列竖直排列的条纹, 其中每一条纹叫做一个全息直线, 这样的全息图叫做只水平视差全息图 HPO (Horizontal-Parallax-Only Holography), 这种消除竖直视差的方法可以使数据量降低1000倍左右。

(2) 减小观察角度。入射光线和衍射光线满足衍射公式:

$$f\lambda = \sin\theta_0 - \sin\theta \quad (1)$$

其中, θ_0 是入射角, θ 是衍射角, λ 是单色光的波长, f 是全息平面上该点的条纹频率。从式(1)中可以看出, 在入射光波长 λ 、入射角 θ_0 一定的情况下, 衍射角 θ 越大, f 就会越大。假设入射角是 0° , 当衍射角接近 90° , 那么 f 为 $1/\lambda$, 根据奈奎斯特采样定理, 最低采样频率应为 $2/\lambda$, 若采用氦氖激光, 采样频率为 $2/(6.328 \times 10^{-7}) \approx 3160$, 也就是每英寸采样点数为 $25.4 \times 3160 = 80264$, 这样不仅数据量大, 而且目前还没有如此高分辨率的输出设备, 减小衍射角, 牺牲部分立体感, 不仅采样数据量变小, 而且降低了对输出设备分辨率的要求, 使得在现有硬件条件下研究 CGH 成为可能^[5,6]。

3. 基于衍射的 HPO 条纹生成法

3.1 空间采样和频率采样

基于衍射的 HPO 条纹生成法忽略了竖直视差, 整个全息平面由变成一系列竖直排列的全息直线组成, 每一条全息直线又被分为一些间隔相同的小线段, 每一小线段称为一个全息元。每个全息元是由频率互不重叠的基本条纹按一定的频率分量线性叠加而成, 所有全息元的条纹组成最终的全息图。

基于衍射的条纹生成方法的核心是对整个条纹在空间上和频率上进行分解。如图1, 对于只有水平视差的全息条纹, 设该条纹的频谱为 $S(s, f)$, x 是全息条纹中的水平位置, f 是在该水平位置处的条纹的空间频率。

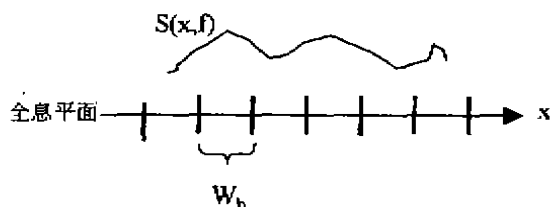


图1 采样

对 $S(x, f)$ 进行采样, 在空间上, 采样间隔 W_k 要足够小, 才能使相邻两小段条纹衍射的光线虽然强度上有差异, 但对人眼来说感觉是平滑过渡的, 人的视觉系统在侧边上能够辨认的最小张角大约一分(290微弧度)。如果观察者离图像距离是600mm, 那么 W_k 应为:

$$W_k = (0.000290) * 600\text{mm} = 0.175\text{mm}$$

在频率空间上, 若宽度为 W_k 的小段内包含有频率为 f_1 和 f_2 的条纹, 它们会将光线衍射向两个不同的方向, f_1 和 f_2 之间的差值 Δf 决定了这两个方向之间的夹角 $\Delta\theta$ 。 $\Delta\theta$ 大到一定的程度, 人眼就会感觉这是两条不同的光线。那么, Δf 最大为多少才不会影响衍射光

线的连续性呢? 根据人眼特性, 人眼的瞳孔直径为3mm, 在观察图像上的某一点时, 人眼瞳孔的张角决定了 Δf 的最大值。对衍射方程两边取导数:

$$\frac{\partial f}{\partial \theta_0} = \frac{\cos \theta_0}{\lambda} \quad (2)$$

如果观察距离为600mm, $\lambda = 0.633\mu\text{m}$, 求得 Δf :

$$\Delta f = (3\text{mm}) / (600\text{mm}) / 0.633\mu\text{m} = 8\text{mm}^{-1}$$

通过按间隔 W_k 对条纹进行空间采样之后, 整个全息平面就由一些全息元组成。全息元有两个重要的性质: (1) 足够小, 看起来是一个点; (2) 一致的频谱, 即全息元是以 W_k 和 Δf 对 $S(x, f)$ 的采样值。

3.2 基本条纹的求法

基本条纹承担朝一定方向衍射光线的任务。对于只有水平视差的情况, 如图2, 每一个基本条纹将光线衍射到一个子区域中去, 其中每个子区域的角度范围是 Δ , 这样, 整个视场在水平方向上就被分为相互不重叠的一些区域。

设一束激光照射到该全息平面上, 将刚透过全息平面上的光的复分布记为 $u(x)$, 距离全息平面距离为 z 的光的分布记为 θ 的函数 $v(\theta)$ 。

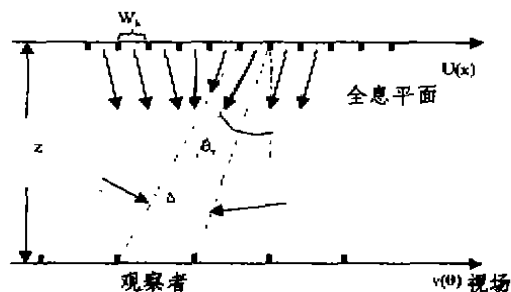


图2 HPO 衍射示意图

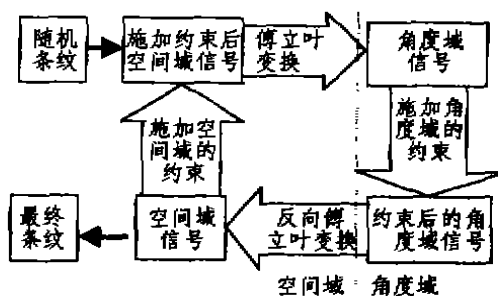


图3 基本条纹生成算法流程

$u(x)$ 和 $v(\theta)$ 之间满足弗朗和费衍射公式:

$$v(\theta) = \frac{e^{ikz(1+\theta^2)}}{ikz} \int_{-\infty}^{\infty} u(x) e^{-i\theta x} dx \quad (3)$$

其中, $k = 2\pi/\lambda$, λ 是入射光的波长。很明显, $v(\theta)$ 和 u

$u(x)$ 是一种傅立叶变换的关系, 可以利用迭代约束法来对信号 $u(x)$ 进行求解^[1]。经过多次循环后, 可以得到最终的 $u(x)$, 即所要求的基本条纹。迭代约束法的过程如图3所示。

4. 一个 CGH 的实现方案

图4为本文的实验方案, 主要包括物像输入、基本条纹生成与叠加、条纹输出与缩小、光学重现等过程。

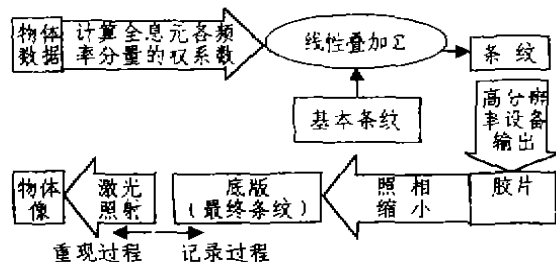


图4 实现过程示意图

本次实验选择波长为632.8nm 氦氖激光作为单色光, 入射角为 10° 。 $W_h = 0.25\text{mm}$, 频域采样间距 $\Delta f = 8\text{mm}^{-1}$, 最后生成的全息条纹输出到 $5\text{mm} \times 7\text{mm}$ 的底版上, 竖直方向上的分辨率为4lp/mm, 共有 $4 \times 7 = 28$ 个点; 水平方向上有 $5/0.25 = 20$ 个全息元, 每个全息元(0.25mm)内采样32个点, 即1mm内取样128个点, 由采样定理知, 条纹的最高频率不超过 64mm^{-1} , 又因为频域采样间距 Δf 为 8mm^{-1} , 所以基本条纹的条数为 $64/8 = 8$ 条。它们所代表的频率范围为 $[i * 8, (i+1) * 8]$, $i = 0, 1, \dots, 7$ 。成像一个点至少需要2条光线的会聚, 所以水平方向上最多采样点数为 $(8/2) \times 20 = 80$ 。本次实验中采用一幅 42×28 的黑白图像作为原像。

采用静态的感光底版记录全息条纹。用3600dpi(黑白分辨率)的激光照排设备模拟输出256灰度级的点, 灰度分辨率达90dpi, 再用照相设备缩小40倍, 达到3600dpi的灰度分辨率。

4.1 生成基本条纹

本次实验中基本条纹的条数为8条, 频率范围为 $[i * 8, (i+1) * 8]$, $i = 0, 1, \dots, 7$, 计算过程如下:

```
for(i=0; i<hBasicEleNumber; i++) // i 为基本条纹的序号,
    hBasicEleNumber 为基本条纹的条数
{
    GenerateRandom(); // 生成随机条纹
    for(j=0; j<100; j++) // 进入迭代约束法, 为迭代的次数
    {
        SpatialFilter(); // 空间滤波
        FourSpatialToAngle(); // 傅立叶正变换
        StoV(); // 从 s(θ) 转移到 v(θ), s(θ) 是 v(θ) 的标准化
        SpectralFilter(); // 进行角度域的滤波
        VtoS(); // 从 v(θ) 转换到 s(θ)
    }
}
```

FourAngleToSpatial(); // 傅立叶反变换

Save(); // 保存得到的基本条纹

其中, SpatialFilter() 执行空间域的滤波, 它所完成的任务就是将各点的复振幅置为1。而 SpectralFilter() 完成角度域的滤波。

在 x 域, $kw = 2\pi/\lambda * W_h = 2482$, 积分的范围可以确定; 而 θ 域的变化范围可以通过衍射公式(1)得到: $\theta_{\max} = \arcsin(f_{\max}/\lambda) = \arcsin(64 * 6.328 * 10^{-4}) = 2.3$ 度

最后得到的基本条纹如图5(只一部分)。

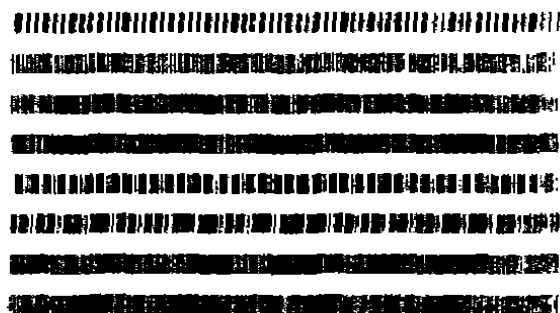


图5 基本条纹图

4.2 全息元向量

基本条纹生成后, 为使全息元完成特定的衍射任务, 基本条纹必须按所需的频率分量进行叠加, 叠加时的权系数构成了全息元向量, 整个全息平面所有全息元向量可记录在一个三维数组 hHoloLine[L][M][N] 中, 其中, L 是全息平面上全息直线条数; M 是一条全息直线上的全息元个数; N 是一个全息元上的基本条纹的个数。具体算法参见图6。

如图6, AB 是全息平面, EB' 是投影平面。要想在点 (x, y, z) (投影后) 处成一个点像, 需如何确定全息元向量才能达到目标? 因为是 HPO 全息图, 所以可以将 y 取整, 作为对竖直条纹的索引。而 z 又是固定的, 所以只看水平方向。亦即看全息平面上的第 y 条全息直线上的各个全息元向量究竟包含有什么样的分量才能使得光线会聚于投影平面上的 (x, y, z) 处。如果全息直线上的第 i 个全息元的第 j 个基本条纹正好将光线衍射到 (x, y, z) 点, 则这个全息元向量相应的第 j 个分量便被置为1。全息直线上的全息元向量的程序如下:

```
for(int i=0; i<hMaxHoloSeg; i++)
{
    for(int j=0; j<hBasicEleNumber; j++)
    {
        if(第 j 个基本条纹能将光线衍射到(x, y, z)点)
            hHoloLine[i][j][y]=1;
    }
}
```

针对物体的每一个数据点,都进行一遍这个操作,最后得到的 hHoloLine 就是这个物体的全息元向量。

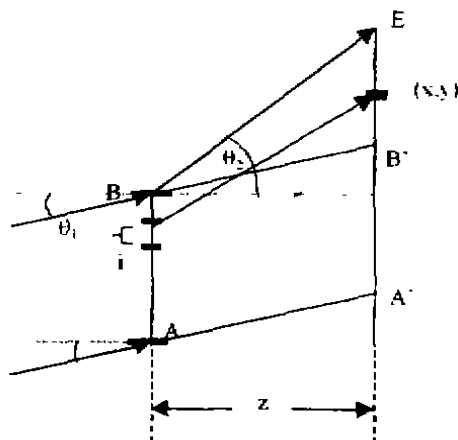


图6 投影俯视图

根据全息元向量,对基本条纹进行加权相加,生成最后的条纹。算法过程如下:

```
for(i=0;i<L;i++) //L 是全息平面上全息直线条数
for(j=0;j<M;j++) //M 是一条全息直线上的全息元个数
{ for(k=0;k<hSampleNumber;k++)
{ total=0;
for(m=0;m<N;m++) //N 是一个全息元上的基本条纹的个数
{ temp=random[m][k];
total+=temp*hHoloLine[i][j][m];
}
fwrite(&total,sizeof(double),1,tempfp); //写入一个暂时文件
} //end for(k)
} //end for(j)
```



图7 程序生成的“王”字的 HPO 全息图

这是一个四重循环,非常耗时,但由于是重复的运

算,便于用硬件实现。将生成的条纹写到一个图像文件中。图7就是本次实验中生成的“王”字的 HPO 全息图。

4.3 光学重现

重现的光路如图8。

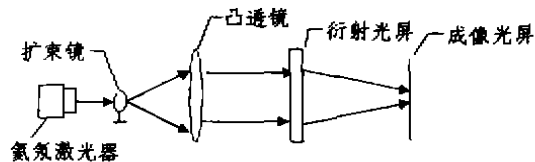


图8 重现光路图

扩束镜将激光器发射的激光进行扩展,凸透镜将发散光变成平行光,衍射光屏是全息条纹所在的平面,光线衍射后会聚到成像光屏,得到像。

结论 计算机生成全息图将计算机技术和全息技术结合在一起,很有可能使半个世纪以前发明的全息技术走向人类的日常生活和科研活动,成为新一代视觉信息的表达手段。可以肯定在下一个世纪初,计算机速度、容量、高分辨率显示器以及传输带宽将能够满足 CGH 的要求。在现有的硬件设备条件下,通过减少数据量的方法能够实现 CGH,为 CGH 实用化研究打下基础。

参考文献

- 1 于美文. 光全息学及其应用. 北京理工大学出版社, 1996. 8
- 2 St. -Hilaire P. et al. Color images with the MIT holographic video display. In: Benton S A, ed., SPIE Vol. 1667, Practical Holography, VI. 1992. 73~84
- 3 Lucente M. Diffraction-Specific Fringe Computation for Electro-Holography: [Ph. D. Thesis]. Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 1994
- 4 Mark L. Interactive three-dimensional holographic displays: seeing the future in depth. Computer Graphics, 1997, 31(2): 63~67
- 5 Hashimoto N. Real-time electroholographic system using liquid crystal television spatial light modulators. Journal of Electronic Imaging, 1993, 2(2): 93~99
- 6 Lee A J, Casasent D P. Computer generated hologram recording using a laser printer. Applied Optics, 1987, 26(1)
- 7 Wyrowski F, Bryngdahl O. Interactive Fourier-transform algorithm applied to computer holography. J. Opt. Soc. Am., 1988, 5(7)