# 双视点 3D 视频文件的裸眼立体组合投影实时显示算法

郭华源[1,2)], 秦开怀[1)*], 毛 苗[1)], 孙 丰[1)]

[1)] (清华大学计算机科学与技术系 北京 100084)
[2)] (中国人民解放军总医院计算机应用与管理科 北京 100853)
(qkh-dcs@tsinghua.edu.cn)

**摘 要:** 基于 Client/Server 结构和 sort-last 并行绘制策略, 提出双视点 3D 视频文件的裸眼立体组合投影实时显示算法. 首先在服务端全屏播放左右或上下格式的 3D 视频文件, 以不低于 25 帧/s 的速率在线截屏并按 JPEG 格式压缩后转发给 12 个 Client PC. 每个 Client PC 接收每一帧截屏图像后, 利用 Fragment Shader 和多渲染目标(MRT)通过一遍绘制完成 2 个单视点子图像的裁剪、缩放、奇偶条纹倾斜绘制, 经几何和亮度校正, 并将 2 个子图像交织后再向前投影到光栅显示屏幕. 该屏幕的投影表面为 3.6 m×1.6 m, 单台投影仪分辨率为 1024×768, 投影系统的分辨率为 3584×1536. 实验结果表明, 该算法的显示帧率≥24 帧/s, 且当条纹倾斜角度为 10°时裸眼立体显示效果最好.

**关键词:** 双视点; 裸眼立体显示; 组合投影; 3D 视频文件
**中图法分类号:** TP391

# Real-Time Tiled Multi-projector Autostereoscopic Display Algorithm for Dual-view 3D Video Files

Guo Huayuan[1,2)], Qin Kaihuai[1)*], Mao Miao[1)], and Sun Feng[1)]

[1)] (*Department of Computer Science & Technology, Tsinghua University, Beijing* 100084)
[2)] (*Department of Computer Application & Management, the General Hospital of PLA, Beijing* 100853)

**Abstract:** To display the dual-view 3D video files, this paper presents a real-time tiled multi-projector autostereoscopic display algorithm based on the Client/Server structure and the sort-last distributed rendering scheme. First, the dual-view 3D video is displayed in a full screen mode on the server. Second, the dual-view images are screen captured at frames per second of more than 25 and distributed to all rendering clients. Then, a GPU-based image processing technique is utilized to split each HSBS or Half-OU image into two single-view sub-images according to the tiled parameters, and to resize the resolution by bilinear interpolation and generate slanted stripe images through a single-pass rendering process. Lastly, after performing the geometric calibration and luminance correction, these images are interleaved to provide the autostereoscopic vision on the optical display screen. The projection system, which is equipped with a front-projection screen that covers an area of 360×160 square centimeters and 24 projectors with a two-dimensional projection resolution of 3584×1536 pixels, can provide glasses-free stereoscopic vision at 24 frames per second. Furthermore, it is verified that the autostereoscopic display reaches the optimum when the slanted degree of stripes is 10°.

收稿日期：2014-05-05; 修回日期：2015-06-03. 郭华源(1979—), 男, 博士研究生, 工程师, 主要研究方向为裸眼立体组合投影显示、图像处理; 秦开怀(1958—), 男, 博士, 教授, 博士生导师, 论文通讯作者, 主要研究方向为计算机图形学及动画、3D 立体显示、小波变换及多分辨率几何造型、超声无损检测; 毛 苗(1987—), 女, 博士研究生, 主要研究方向为 3D 绘制、立体匹配; 孙 丰(1983—), 男, 博士研究生, 主要研究方向为立体绘制.

**Key words:** dual-view; glasses-free; autostereoscopic display; tiled multi-projector; 3D video files

## 1　相关工作

裸眼立体(autostereoscopic)[1-2]组合投影显示技术是指不需佩戴立体眼镜等设备，凭双眼可直接从多个投影仪无缝拼接的投影图像获得立体视觉效果的一种 3D 显示技术，它是目前 3D 显示领域的研究热点之一．裸眼立体组合投影系统致力于生成大屏幕、高分辨率、沉浸感强的立体显示效果，可广泛应用于远程医疗、虚拟现实、娱乐影院、科学展览等领域．

双视点 3D 视频文件是一种广泛应用的立体影像多媒体数据存储和交换方式．厂商生产的 3D 视频文件常以 25 帧/s 或 30 帧/s 速率将双视点图像以左右(HSBS)或上下(Half-OU)格式保存，如图 1 所示．其中，左右格式能使图像垂直方向的分辨率不变，水平方向的分辨率减半；而上下格式则使图像水平方向的分辨率不变，垂直方向的分辨率减半．

a. 左右格式　　b. 上下格式

图 1　双视点 3D 视频图像

目前，国内外有关双视点 3D 视频文件裸眼立体组合投影显示的研究成果还未见报道．本文对此问题进行深入分析，主要贡献包括：1) 提出一种双视点 3D 视频文件的裸眼立体组合投影实时显示算法，获得良好的立体显示效果和实时性能，投影屏幕为 3.6 m×1.6 m，数字分辨率为 3584×1536，显示帧数为≥24帧/s; 2) 对不同奇偶条纹倾斜角度下的双视点裸眼立体显示效果进行比较分析．

## 2　研究现状

### 2.1　多视图绘制

常见的 3D 数据可分为 2 种来源，计算机图形学绘制和摄像机拍摄；其中，计算机图形学绘制通常采用多视图多遍绘制的方法[3]，但是其耗时与视点个数成正比，难以完成多视点图像实时绘制．

近年来出现一些基于 GPU 加速的绘制方法，用于解决多视点数据多遍绘制速度慢的问题，如多视图点 splatting 法[4]、单遍多视图体绘制法[5]等．Schmidt 等[6]通过 8 遍绘制将 8 个视点图像绘制到不同纹理，在第 9 遍绘制时进行交织显示．Kooima 等[7]使用 GPU 可编程 Vertex 和 Fragment 流水线部件通过一遍绘制实现多视点图像的交织，获得实时交互绘制效果．Marco 等[8]基于全息显示屏幕提出一种 GPU 加速的体光线投射方法，通过 2 遍绘制对复杂体数据集进行交互操作，并获得真实感效果．Sorbier 等[9]使用 Geometry shader 基于多视点图像绘制立体场景，通过一遍绘制技术大幅提高绘制帧率．

此外，WireGL[10]，Chromium[11]和 AnyGL[12]等分布式图形绘制系统支持 sort-first 或 sort-last 并行绘制，采用流处理单元拦截和修改 OpenGL 指令，使现有 OpenGL 应用软件支持大屏幕拼接绘制．

### 2.2　实时视频处理

对摄像机拍摄的视频数据进行实时显示是计算密集型任务．Park 等[1]提出一个投影式多视点裸眼立体显示系统，该系统由 4 台摄像机、1 台计算机和 4 台投影仪组成，能够以 15 帧/s 速率显示 4 个视点的 VGA 图像，基本满足实时性应用需求．Shin 等[13]根据 3D 实景图像和深度图像，通过视觉合成技术生成 8 个虚视点图像，该算法使用 GPU 加速后获得 12 倍加速比，可用于实时显示．

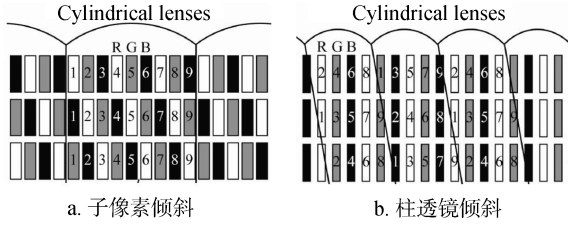
Daniel 等[14]实现一个用于医学图像绘制的裸眼立体显示系统 iGlance．该系统使用 H.264 标准对 4 个分辨率均为 1280×720 的视点图像进行编码传输；然后，接收端使用自由视点生成技术在原有 4 个视点基础上新增 5 个视点图像；最后，9 个视点图像交织绘制成一幅分辨率为 3840×2160 的图像，完整显示在 1 台 QuadHD LCD 显示器上．

### 2.3　立体感增强

在立体感增强方面，主要采用多视点图像叠加[15]、多视点图像条纹交织[16]和倾斜条纹交织[17]等技术．条纹交织方式分为行交织、列交织等．其中，行交织用于液晶、阴极电子射线管等显示器件，列交织用于投影显示．根据文献[18-19]可知，当显示分辨率一定时，视点图像分辨率和视点个数成反比．

因此，对多视点图像交织绘制时，将不同视点图像的 RGB 子像素条纹在垂直方向上进行小角度

倾斜[19]，使图像分辨率的损耗分摊到水平和垂直 2 个方向上，可优化立体显示效果，如图 2a 所示. 同种子像素在垂直方向依次右移 1 个子像素宽度，倾斜角 $\phi = \arctan(1/3) = 18.44^\circ$. 这种子像素条纹倾斜要求显示器件工艺精度高，如液晶显示器.




a. 子像素倾斜 b. 柱透镜倾斜

图 2 2 种倾斜绘制 RGB 条纹图像的方法

文献[18]详细分析将彩色显示器中的透镜阵列倾斜放置，可增强多视点图像裸眼立体显示效果的原理，如图 2b 所示，并给出倾斜角的计算公式 $\phi = \arctan(H/(V \times \gamma))$. 其中，$H$ 和 $V$ 分别是水平和垂直向像素点的长度，$\gamma$ 是常数，通常 $\gamma = 2$. 但由于这种柱透镜中心线的倾斜[20]，导致观察者在垂直向移动时观看到的图像在水平向发生旋转.

另外，透镜光栅物理上的局限性使子像素的边界不能准确对齐，从而导致串扰. 王琼华等[20]分析显示图像子像素与柱透镜光栅之间的位置关系，提出一种通过调整显示图像子像素的位置来减少串扰现象的方法.

本文实验室的光学屏幕由柱透镜光栅、有机玻璃硬幕和漫反射软幕组成[21]，具有各向异性的反射性质. 投影光线经柱透镜光栅反射和折射形成图像，难以达到子像素级精度. 所以，本文采用像素倾斜技术[17]进行多视点图像交织，提高立体显示效果.

### 2.4 裸眼立体投影显示系统

2010 年，作者所在实验室成功研制一套由 24 台投影仪组成的多投影仪自由立体显示系统[21]，提出几何校正两步法和自适应多模板亮度校正法[22]解决无缝拼接和亮度平滑问题，开发基于 Chromium 的非侵入式多视图并行绘制系统[17]，使传统的 OpenGL 应用程序(游戏、动画等)无需修改代码即能进行空间复用的自由立体投影显示，获得良好的裸眼立体显示效果. 但该系统还存在以下不足：

1) 不支持双视点 3D 视频文件的裸眼立体显示;

2) 投影清晰度有待提高，立体效果仍需加强.

因此，本文研制基于 GPU 的快速算法，实现双视点 3D 视频文件的裸眼立体组合投影实时显示;同时，寻找适当的奇偶条纹倾斜角度，使图像清晰度和立体效果趋于最好.

## 3 系统结构

本文借鉴桌面拷屏分发的思路[22]，提出一种面向双视点 3D 视频文件的裸眼立体显示算法. 该系统采用主从式结构，由 1 个服务端, 12 个绘制客户端, 24 台投影仪和光学投影屏幕组成. 系统结构和算法流程分别如图 3~4 所示.

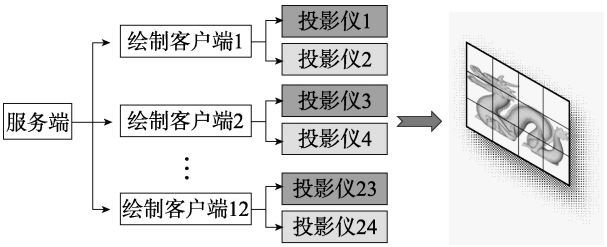



图 3 裸眼立体组合投影显示系统结构图

首先，服务端使用 Windows Media Player 等播放器全屏播放 3D 视频文件，然后在线截屏并经 JPEG 压缩后转发 12 个客户端. 每个客户端接收到 3D 视频流的每一幅图像后，都要先根据组合投影参数对它们进行裁剪、缩放、条纹绘制、几何与亮度校正等处理;然后以 Windows 双屏扩展的方式投影输出. 其中，编号为 1, 3, …, 23 等 12 个投影仪拼接生成左视点图像，而剩余投影仪拼接生成右视点图像. 最后，左右视差图像交织显示在由透镜光栅阵列构成的光学屏幕上，产生裸眼立体显示效果. 将上述过程循环执行，直至结束投影显示.

本文系统采用 sort-last 并行绘制策略，将图像数据平均分配给各个客户端独立绘制，并各自将结果投影输出合成立体显示图像. 因而，任务划分简单、扩展性好、易于负载平衡.

其中，客户端的计算任务包括 4 个子过程，分别对应图 4 中的第 6~9 步：1) 根据组合投影参数从解压后图像中裁剪出该客户端对应的 2 个单视点子图像，即为该客户端的有效投影显示区域; 2) 根据投影仪的分辨率，经双线性插值分别调整 2 个单视点子图像的分辨率; 3) 对 2 个单视点子图像进行奇偶条纹倾斜交织绘制; 4) 对 2 个单视点子图像进行几何校正和亮度校正，使拼接投影图像的几何形状一致，亮度变化连续.

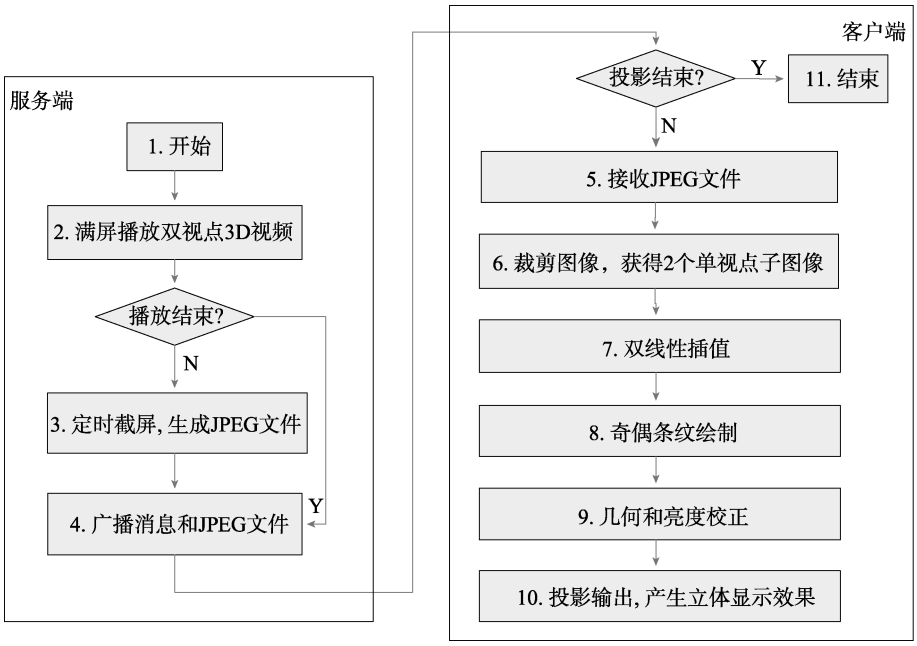



图 4　裸眼立体组合投影显示算法流程

## 4　视频图像处理

### 4.1　图像裁剪

根据左右或上下格式的 3D 视频图像特点，经过裁剪得到左右 2 个单视点的子图像. 3D 视频图像文件是按照从上至下、从左至右顺序保存每个像素点的 RGB 3 个颜色分量值.

本文将图像裁剪问题转化为像素点在 GPU 中的并行移位处理，实现快速切分. 如图 5a 所示，对于左右格式截屏图像，假设其分辨率是 1600×900，则左右单视点图像的分辨率都是 (1600/2)×900=800×900，即水平方向左起第 1~800 列为左视点图像，第 801~1600 列为右视点图像. 然后，根据客户端组合投影参数进行裁剪，分别获得 2 个单视点子图像，如图 5b 中黑灰色矩形所示. 同理，对上下格式截屏图像也进行类似处理.

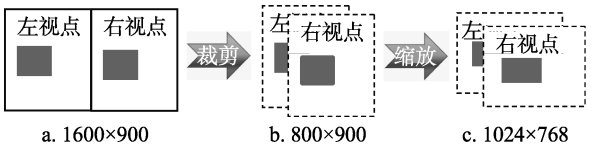



图 5　左右格式截屏图像的裁剪与缩放处理

### 4.2　图像缩放

通常裁剪出的单视点图像分辨率与单台投影仪分辨率不一致. 对于左右格式，如图 5b 所示，在水平方向上，图像分辨率小于投影分辨率(800 < 1024)，需增加像素点；而在垂直方向上，图像分辨率大于投影分辨率(900>768)，需减少像素点. 因此，采用双线性插值进行缩放，结果如图 5c 所示.

显然，上述方法优点是投影图像轮廓完整；满屏投影，拼接投影显示面积大；对不同分辨率图像，不需要改变几何校正和亮度校正参数. 但缺点是计算量大，必须采用加速算法来确保实时性.

### 4.3　双视点图像交织

多视点图像交织方式是影响立体显示效果的重要因素. Michael 等[23]通过对双视点图像进行棋盘格交织，同时配合立体眼镜的开闭确保观众获得正确的深度信息，从而增强立体显示效果. 文献[17]采用多视点图像倾斜交织绘制来提高立体感.

通常使用模板技术实现多视图交织[4,17]，即先根据交织方式生成模板，然后进行多视图交织. 定义二维模板数组 *mask*，其中 *mask*[*i*][*j*]表明交织图像第 *i* 行第 *j* 列的像素值是来自编号为 *mask*[*i*][*j*] 的视点图像. 为加快绘制速度，可将模板数据转换为纹理坐标[17]，从而把多视点图像交织问题转变成根据模板的纹理坐标值对多视图纹理进行采样.

本文提出双视点图像奇偶条纹倾斜交织方法，通过调整条纹倾斜角度，优化投影显示效果，如图 6 所示. 其中，L 和 R 分别表示左右视图像素点，$P_i$ 表示第 *i* 列像素(*i*=1, 2, 3, …)；$\phi$ 为条纹倾斜角，当其为 45°时，即为棋盘格交织，如图 7 所示.

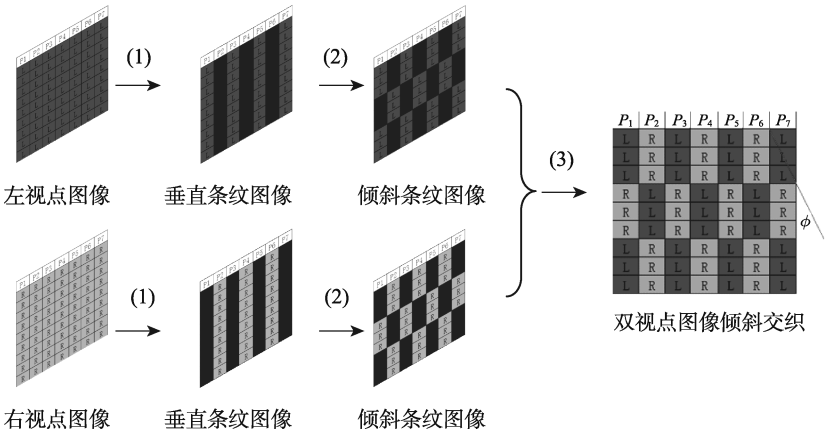



图 6 双视点图像奇偶条纹倾斜交织示意图

| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ |
|---|---|---|---|---|---|---|
| L | R | L | R | L | R | L |
| R | L | R | L | R | L | R |
| L | R | L | R | L | R | L |
| R | L | R | L | R | L | R |
| L | R | L | R | L | R | L |
| R | L | R | L | R | L | R |
| L | R | L | R | L | R | L |
| R | L | R | L | R | L | R |
| L | R | L | R | L | R | L |

图 7 棋盘格交织

双视点图像奇偶条纹倾斜交织绘制原理如下：

Step1. 对左右视点图像分别生成列奇偶相间垂直条纹图像.

Step2. 引入错切矩阵，将列奇偶相间垂直条纹图像变换为奇偶相间倾斜条纹图像.

Step3. 双视点倾斜条纹图像同时投影在屏幕上,生成奇偶相间倾斜条纹交织图像.

根据错切变换定义，给定错切角度 $\phi$(即为图 6 倾斜角 $\phi$)，则平面上点 $(x,y)$ 错切变换为点 $(x+y\tan\phi, y)$ 用矩阵表示为

$$\begin{pmatrix}1 & \tan\phi\\ 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix}=\begin{pmatrix}x+y\tan\phi\\ y\end{pmatrix}.$$

经观察发现，奇(偶)条纹是相间排列，条纹宽度为 1，条纹高度 $\rho$ 由错切角度 $\phi$ 决定，即

$$\rho = floor(1.0/\tan\phi).$$

函数 $floor(x)$ 指计算不大于的最大整数. 因为像素点的原子性，条纹高度值最小为 1(错切角度为 45°)，最大为正无穷(错切角度为 0°)；单视点图像中水平向相邻 2 条纹像素 $x$ 坐标值的奇偶性相反，但其条纹高度应相等.

因此，取条纹高度值作为倍长，计算每个像素点的 $y$ 坐标值与倍长值的商，然后利用像素点 $x$ 坐标值奇偶性与商值奇偶性之间的对应关系，生成奇偶条纹. 条纹倾斜角度的变化范围为[−45°，45°]，但是双视点条纹在水平方向具有对称性，所以条纹倾斜角度的等效变化范围为[0°，45°]. 使用倍长法，由左右格式 3D 图像生成奇偶相间倾斜条纹图像的核心代码如下(GLSL 语言):

```
//左视点图：置偶数列像素值为 0(黑)
if(mod(floor(pos.x),2.0)==mod(floor(pos.y/ρ),2.0))
    vec4Result = vec4(0.0, 0.0, 0.0, 1.0);
//右视点图：置奇数列像素值为 0(黑)
if(mod(floor(pos.x),2.0)!=mod(floor(pos.y/ρ),2.0))
    vec4Result = vec4(0.0, 0.0, 0.0, 1.0);
```

## 5 加速算法

双视点图像实时绘制是本文算法的关键.针对绘制场景，文献[9]采用帧缓存对象(frame buffer object，FBO)和多渲染目标(multiple render target，MRT)技术，完成多个视点图像实时绘制. MRT 的特点是采用 FBO 与多个颜色纹理关联，将多个颜色缓存渲染到多个 FBO 关联存储区. 本文将图像裁剪、缩放、奇偶条纹倾斜绘制等计算密集型任务放到 GPU 中执行，基于 MRT 技术实现一遍绘制双视点子图像，以保证裸眼立体显示的实时性.

算法 **1.** 基于 MRT 的双视点子图像绘制算法

输入. 双视点图像、子图像裁剪起止坐标、缩放和倍长因子.

输出. 双视点渲染结果子图像.

Step1. while(客户端接收一幅双视点截屏图像){

Step2. 使用 PBO 技术将截屏图像从内存传入 GPU

//绘制左视点子图像.

Step3. 将渲染缓存图像关联到 FBO 准备离屏渲染.

Step4. 启用片断着色器处理程序对象.

Step5. 向片断着色器处理程序传递左视点子图像的裁剪、缩放、条纹倍长等参数.

Step6. 设置渲染对象为 GL_COLOR_ATTACHMENT0_EXT.

Step7. 进行离屏渲染, 实现同步绘制.

Step8. 停用片断着色器处理程序对象.

Step9. 使用 PBO 技术, 从 GL_COLOR_ATTACHMENT0_EXT 中获取左视点子图像.

//绘制右视点子图像

Step10.将渲染缓存图像关联到 FBO 准备离屏渲染.

Step11. 启用片断着色器处理程序对象.

Step12. 向片断着色器处理程序传递右视点子图像的裁剪、缩放、条纹倍长等参数.

Step13. 设置渲染对象为 GL_COLOR_ATTACHMENT1_EXT.

Step14. 进行离屏渲染, 实现同步绘制.

Step15. 停用片断着色器处理程序对象.

Step16. 使用 PBO 技术, 从 GL_COLOR_ATTACHMENT1_EXT 中获取右视点子图像.

}

算法 1 有 3 点注意事项：

1) 视频图像处理需要在主机内存和 GPU 之间频繁传输数据, 所以数据传输速度成为影响算法性能的一个关键因素. 像素缓存对象(pixel buffer object, PBO)使用直接内存访问技术实现帧缓存和主机内存之间数据高速传输. 因此, 算法 1 在输入截屏图像和输出渲染结果时都使用 PBO 技术, 这是本文算法与文献[9]方法的不同之处.

2) 片断着色器处理程序负责双视点截屏图像的裁剪、缩放和倾斜条纹绘制. 由于左右格式和上下格式双视点截屏图像的裁剪原理不同, 运行算法 1 之前就需要根据双视点 3D 视频格式选择相应的片断着色器处理程序.

算法 **2.** 左右格式图像片断着色器处理算法

输入. 视点值、子图像裁剪坐标、缩放和倍长因子.

输出. 双视点渲染结果子图像.

Step1. 左视点：

Step1.1. 水平移位从图像左半部裁剪左子视图;

Step1.2. 双线性插值, 进行图像缩放;

Step1.3. 用倍长法生成左视点倾斜条纹子图像.

Step2. 右视点：

Step2.1. 水平移位从图像右半部裁剪右子视图;

Step2.2. 双线性插值, 进行图像缩放;

Step2.3. 用倍长法生成右视点倾斜条纹子图像.

算法 **3.** 上下格式图像片断着色器处理算法

输入. 视点值、子图像裁剪坐标、缩放和倍长因子.

输出. 双视点渲染结果子图像.

Step1. 左视点:

Step1.1. 垂直移位从图像上半部裁剪左子视图;

Step1.2. 双线性插值, 进行图像缩放;

Step1.3. 用倍长法生成左视点倾斜条纹子图像.

Step2. 右视点:

Step2.1. 垂直移位从图像下半部裁剪右子视图;

Step2.2. 双线性插值, 进行图像缩放;

Step2.3. 用倍长法生成右视点倾斜条纹子图像.

3) 经 GPU 渲染后每个客户端中每一个单视点倾斜条纹子图像都需进行几何和亮度校正, 才能保证投影显示具有无缝拼接和亮度平滑的特性. 因此, 本文使用“两步法”和“自适应多模板亮度校正法”[22]分别计算每个客户端关于两个视点图像的几何和亮度校正参数. 只要投影仪的空间位姿和投影亮度等无明显改变, 参数将长期适用.

## 6 测试分析

裸眼立体组合投影显示系统采用前投方式, 共提供两个视点, 每个视点由 3(行)×4(列)个投影仪组成. 光学投影屏幕为 3.6 m×1.6 m, 数字分辨率为 3584×1536. 测试环境如表 1 所示.

**表 1**　实验测试环境

| 名称 | 硬件配置 |
|---|---|
| 服务端 | Intel(R) Core(TM)i7-3520M CPU 2.90 GHz, 4.0 GB RAM |
| 客户端 | Intel(R) Core(TM)2 Duo CPU 2.66 GHz, 2.0 GB RAM, GeForce 9600GT |
| 投影仪 | PLUS U5-762h, 分辨率 1024×768 |

### 6.1 立体显示效果

为分析奇偶条纹倾斜角对投影显示效果的影响, 将倾斜角分别赋值 0°, 5°, 10°, 12°, 15°, 20°, 45°; 然后使用本文算法对 3D 视频文件进行投影显示, 并选择 5 个不同场景图像作对比, 其左右视点投影图像与原图像的峰值信噪比(peak signal to noise ratio, PSNR)曲线如图 8 所示. 为排除 JPEG 压缩所致的图像质量下降, 本文分别将不同倾斜角度的投影图像与 JPEG 解压图像进行了比较. 结果发现, 随着条纹倾斜角度的增大, 左右视点图像质量总体呈现逐渐降低的趋势. 原因是莫尔效应随着倾斜角增大而逐渐增强, 导致图像噪声增大, 清晰度下降.

光学屏幕表层的柱透镜光栅(采用 Microlen 公司 3D15 立体光栅)是竖直安装, 投影图像的倾斜条纹与光栅产生莫尔条纹[24], 相关参数见表 2.

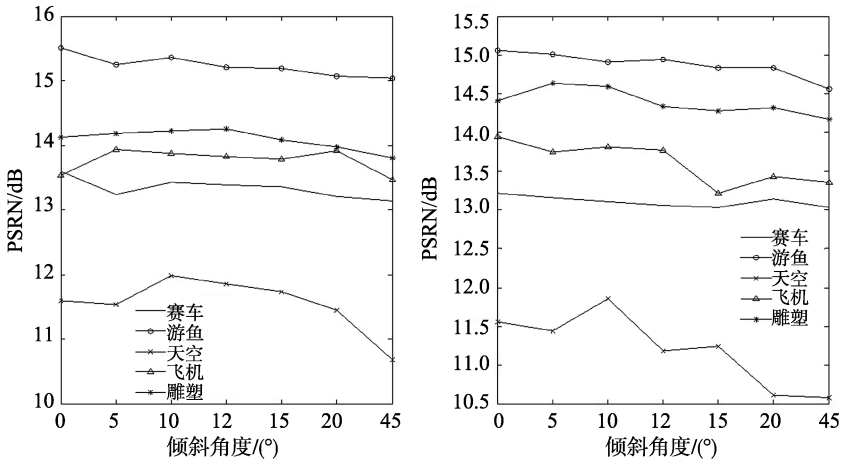



图 8 PSNR 值曲线

表 2 光学屏幕系统相关参数

| 参数名称 | 数值 |
|---|---|
| 柱透镜光栅的节距 $a$/mm | 1.69 |
| 图像条纹宽度 $b$/mm | 1.00 |
| 可分辨最小莫尔条纹宽度 $P$/mm | 3.00 |
| $n$ 为 $a/b$ 附近的正整数 | 2 |
| 最大视角/(°) | 22 |
| 建议观看距离/mm | 3 000 |

根据文献[24]中公式(7.2)知，当 $a$ 和 $b$ 值一定时，若希望人眼看不到莫尔条纹，则光栅与图像条纹的倾斜角 $\phi$ 应满足下列关系

$$\phi < \arccos\frac{P^2\left(n^2a^2+b^2\right)-a^2b^2}{2nabP^2} \approx 15.64^\circ$$

或

$$\phi > \arccos\frac{a^2b^2-P^2\left(n^2a^2+b^2\right)}{2nabP^2} \approx 164.36^\circ \quad (1)$$

为测试不同倾斜角度下莫尔条纹的强度变化，本文对左右视点都采用白色图像进行投影显示. 结果发现当 $\phi=10^\circ$ 时，莫尔效应最小；当 $\phi>15^\circ$ 时莫尔条纹明显增多，正好与式(1)一致. 图 9 所示为屏幕中下方区域莫尔条纹变化情况.

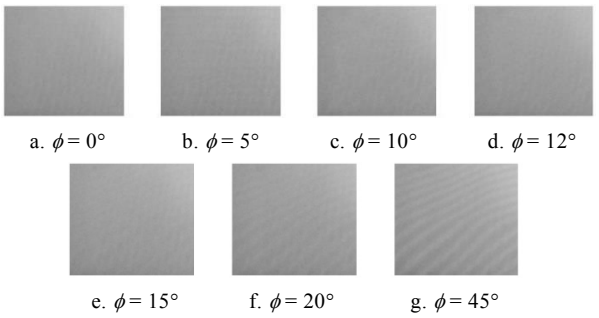



图 9 不同倾斜角下屏幕莫尔条纹的变化

本文采用主观方式评价投影显示的立体效果. 评价人员由 4 男 3 女组成，年龄分布在 22~35 岁之间，均具有正常的立体视觉功能. 测试分成两批，每批 3~4 人，观看点距离投影屏幕为 3.2 m. 每一批测试分为 2 个环节，分别观看 2 部不同内容的 3D 视频，播放顺序如表 3 所示.

表 3 播放顺序

| 内容 | 视频 1 | 间歇 | 视频 1 | 间歇 | … | 视频 1 | 间歇 | 视频 2 | 间歇 | … | 视频 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 倾斜角/(°) | 0 | | 5 | | | 45 | | 0 | | | 45 |
| 时长/min | 2 | 2 | 2 | 2 | … | 2 | 2 | 2 | 2 | … | 2 |

评价内容包括图像清晰度、莫尔效应、立体感. 评价分为优秀、良好、一般、差、很差五个等级，分值依次为 5, 4, 3, 2, 1. 评价结果如表 4 所示.

表 4 立体感主观评价统计表

| 倾斜角/(°) | 图像清晰度 | 莫尔效应 | 立体感 |
|---|---|---|---|
| 0 | 3.27 | 3.05 | 3.04 |
| 5 | 3.02 | 3.28 | 3.16 |
| 10 | 3.43 | 3.51 | 3.59 |
| 12 | 3.25 | 3.42 | 3.57 |
| 15 | 3.45 | 3.11 | 3.46 |
| 20 | 3.26 | 3.23 | 3.36 |
| 45 | 3.01 | 2.4 | 3.16 |

由表 4 可知，当倾斜角度为 10°~15°时，图像清晰度较高，莫尔条纹干扰相对较少，立体感强. 这与图 9 基本一致，与图 8 稍有出入. 原因是 PSNR 值是基于对应像素点间误差的图像客观评价指标，并未考虑人眼的视觉特性.

综上，在本文实验中，当倾斜角为 10°时裸眼立体投影显示效果最好. 图 10 所示为上下格式 3D 视频在 10°倾斜角的裸眼立体组合投影显示效果.

图 10　裸眼立体组合投影显示

另外，本文与 overlapping 方式比较，由于没有进行奇偶交织，overlapping 投影图像的亮度明显增强；但又产生 2 个负面效应：1) 画面颜色失真，局部区域出现高光；2) 在某些视角范围内有辐条状条纹干扰，明显影响视觉效果，如图 11 所示. 采用奇偶条纹倾斜交织投影，尽管图像亮度降低，也有莫尔条纹，但画面层次丰富色彩逼真，亮度平滑连续，深度感较强，立体视觉效果要好一些. 同时，10°倾斜角的视觉效果要优于 45°倾斜角.

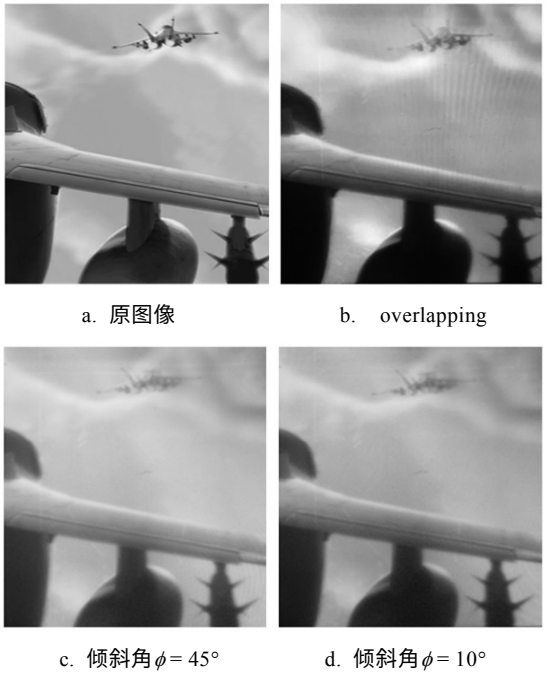



图 11　投影图像质量比较

### 6.2　GPU 加速性能

将 GPU 加速算法与 CPU 串行算法进行性能比较，如图 12 所示. 首先，由于 GPU 的高度并行性，当截屏图像转发频率为 25 帧/s 时，GPU 并行算法能够达到 21.8 帧/s 的绘制帧率，并行加速比为 9.3，这就显著提高双视点 3D 视频文件组合投影显示速度，使投影画面完整、亮度均匀、播放流畅. 其次，随着截屏图像分辨率的提高，GPU 中参与运算的可编程流水线部件也明显增多，导致同步开销增大，使得绘制帧率和加速比逐渐下降；但是与左右格式图像相比，上下格式图像处理性能的降幅要小些. 最后，随着截屏图像转发频率的加快，两种格式图像的绘制帧率都有变化，例如当转发频率为 35 帧/s 时，分辨率为 1280×800 的上下格式图像绘制帧率达到 24 帧/s，加速比为 10.7.

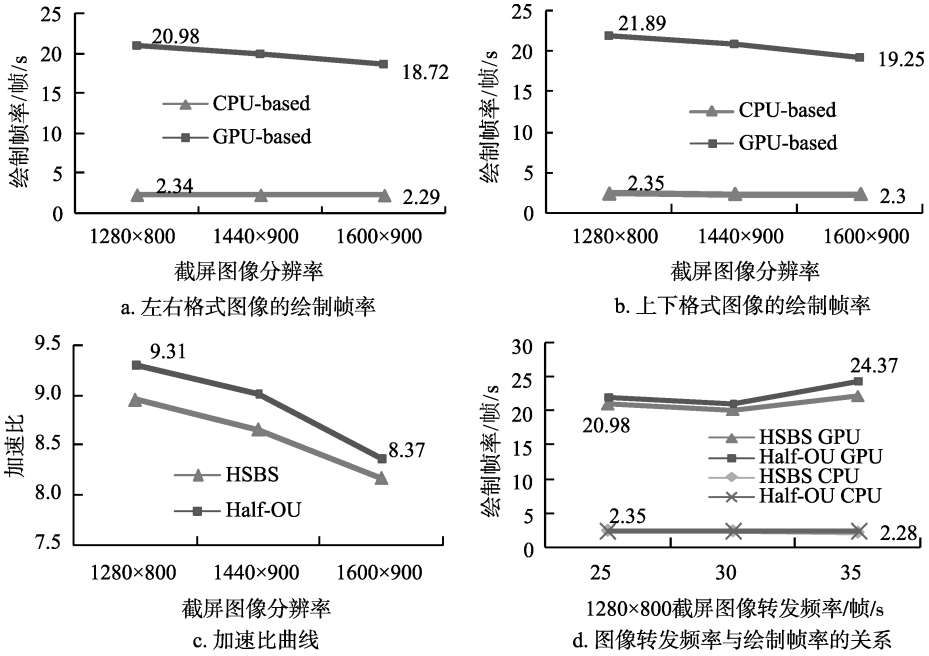



图 12　GPU 算法加速性能

## 7 结 语

本文提出了一种双视点 3D 视频文件的裸眼立体投影实时显示算法，能够获得良好的立体显示效果和实时性能. 并且，对不同奇偶条纹倾斜角度下的双视点裸眼立体显示效果(图 10 和图 11 是照相机拍摄的结果，因此它不如裸眼观察的 3D 效果好)进行比较，发现当倾斜角为 10°时，裸眼立体组合投影显示效果最好. 诚然，根据条纹倾斜交织原理可知，这是近似最优解. 但本文揭示出不同倾斜角度对立体投影显示效果的影响特点有助于我们进一步掌握立体投影显示规律，不断提高裸眼 3D 显示效果.

## 参考文献(References):


[1] Park Y G, Kim S C, Lee S T, *et al*. Implementation of projection-type autostereoscopic multiview 3D display system for real-time applications[C] //Proceedings of SPIE. Bellingham: Society of Photo-Optical Instrumentation Engineers, 2004, 5291: 245-254

[2] Qin Kaihuai, Luo Jianli. Techniques for autostereoscopic display and its development[J]. Journal of Image and Graphics, 2009, 14(10): 1934-1941(in Chinese)
(秦开怀，罗建利. 自由立体显示技术及其发展[J]. 中国图象图形学报, 2009, 14(10): 1934-1941)

[3] Perlin K, Poultney C, Kollin J S, *et al*. Recent advances in the NYU autostereoscopic display[C] //Proceedings of SPIE. Bellingham: Society of Photo-Optical Instrumentation Engineers, 2001, 4297: 196-203

[4] Hubner T, Zhang Y C, Pajarola R. Multi-view point splatting[C] //Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia. New York, ACM Press, 2006: 285-294

[5] Hubner T, Pajarola R. Single-pass multi-view volume rendering[C] //Proceedings of IADIS International Conference Computer Graphics and Visualization. Lisbon: Portugal, 2007: 1-9

[6] Schmidt A, Grasnick A. Multiview point autostereoscopic dispays from 4D-Vision GmbH[C] //Proceedings of SPIE. Bellingham: Society of Photo-Optical Instrumentation Engineers, 2002, 4660: 212-221

[7] Kooima R L, Peterka T, Girado J I, *et al*. A GPU Sub-pixel algorithm for autostereoscopic virtual reality[C] //Proceedings of IEEE Virtual Reality Conference. Los Alamitos: IEEE Computer Society Press, 2007: 131-137

[8] Agus M, Gobbetti E, Antonio J, *et al*. GPU accelerated direct volume rendering on an interactive light field display[J]. Computer Graphics Forum, 2008, 27(2): 231-240

[9] de Sorbier F, Nozick V, Biri V. GPU rendering for autostereoscopic displays[C] //Proceedings of the 4th International Symposium on 3D Data Processing, Visualization and Trans mission. Atlanta: Georgia Instiute of Technology Press, 2008: 1-7

[10] Humphreys G, Buck I, Eldridge M, *et al*, Distributed rendering for scalable displays[C] //Proceedings of IEEE Conference on Supercomputing. Los Alamitos: IEEE Computer Society Press, 2000: 30-37

[11] Humphreys G, Houston M, Ng R, *et al*. Chromium: a stream-processing framework for interactive rendering on clusters[J]. ACM Transactions on Graphics, 2002, 21(3): 693-702

[12] Yang J, Shi J Y, Jin Z F, *et al*. Design and implementation of a large-scale hybrid distributed graphics system[C] //Proceedings of the 4th Eurographics Workshop on Parallel Graphics and Visualization. Aire-la-Ville: Eurographics Association Press, 2002: 39- 49

[13] Shin H C, Kim Y J, Park H, *et al*. Fast view synthesis using GPU for 3D display[J]. IEEE Transactions on Consumer Electronics, 2008, 54 (4): 2068-2076

[14] Ruijters D, Zinger S. IGLANCE: transmission to medical high definition autostereoscopic displays[C] //Proceedings of the True Vision-Capture, Transmission and Display of 3D Video. Los Alamitos: IEEE Computer Society Press, 2009: 1-4

[15] Jaynes C, Ramakrishnan D. Super-resolution composition in multi-projector displays[C] //Proceedings of IEEE International Workshop on Projector-Camera Systems. Los Alamitos: IEEE Computer Society Press, 2003: 354-356

[16] Keller A, Heidrich W. Interleaved sampling[C] //Proceedings of the 12th Eurographics Workshop on Rendering Techniques. Heidelberg: Springer, 2001: 269-276

[17] Luo J L, Qin K H, Zhou Y X, *et al*. GPU-based multi-view rendering for spatial-multiplex autostereoscopic displays[C] //Proceedings of the 3rd IEEE ICCSIT, International Conference on Computer Science and Information Technology. Los Alamitos: IEEE Computer Society Press, 2010: 28-32

[18] Clarkle J A, van Berkel C. Autostereoscopic display apparatus: USA, 6064424[P]. 2000-05-16

[19] Takaki Y, Yokoyama O, Hamagishi G. Flat panel display with slanted pixel arrangement for 16-view display[C] //Proceedings of SPIE. Bellingham: Society of Photo-Optical Instrumentation Engineers, 2009, 7237: 723708

[20] Wang Q H, Li X F, Zhou L, *et al*. Cross-talk reduction by correcting the subpixel position in a multiview autostereoscopic three-dimensional display based on a lenticular sheet[J]. Applied Optics, 2011, 50(7): B1-B5

[21] Luo J L, Qin K H, Zhou Y X, *et al*. GPU rendering for tiled multi-projector autostereoscopic display based on chromium[J]. The Visual Computer, 2010, 26(6-8): 457-465

[22] Zhou Yanxia. Research on geometric and photometric calibrations for multi-projector autostereoscopic display[D]. Beijing: Tsinghua University, 2011: 28-29(in Chinese)
(周艳霞. 多投影仪自由立体显示的几何和亮度校正技术研究[D] . 北京: 清华大学. 2011: 28-29)

[23] McCormick M D, Neal H W, Hutchison D C. Implementation of stereoscopic and dualview images on a micro-display high definition television[C] //Proceedings of 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video. Los Alamitos: IEEE Computer Society Press, 2008: 33-36

[24] Wang Qionghua. 3D display technology and device[M]. Beijing: Science Press, 2011: 106-108 (in Chinese)
(王琼华. 3D 显示技术与器件[M]. 北京: 科学出版社, 2011: 106-108)