

# 虚拟场景中基于 Motion Vector 的人群图像绘制算法

霍奕<sup>1,2</sup>,王喜年<sup>1</sup>,张瑜<sup>3</sup>,陈敬利<sup>4</sup>,马海滨<sup>1</sup>,张鹏飞<sup>5</sup>

HUO Yi<sup>1,2</sup>,WANG Xi-nian<sup>1</sup>,ZHANG Yu<sup>3</sup>,CHEN Jing-li<sup>4</sup>,MA Hai-bin<sup>1</sup>,ZHANG Peng-fei<sup>5</sup>

1.河北师范大学 职业技术学院,石家庄 050031

2.北京工业大学 计算机学院,北京 100022

3.河北科技大学 高教研究所,石家庄 050018

4.河北师范大学 物理科学与信息工程学院,石家庄 050031

5.石家庄市公安局,石家庄 050091

1.College of Career Technology,Hebei Normal University,Shijiazhuang 050031,China

2.College of Computer Science and Technology,Beijing University of Technology,Beijing 100022,China

3.Institute of Higher Education,Hebei University of Science and Technology,Shijiazhuang 050018,China

4.College of Physics Science and Information Engineering,Hebei Normal University,Shijiazhuang 050031,China

5.Shijiazhuang Municipal Public Security Bureau,Shijiazhuang 050091,China

E-mail:emma200604@hotmail.com

**HUO Yi,WANG Xi-nian,ZHANG Yu,et al.People-crowd rendering inside virtual scene by image-based rendering algorithm using Motion Vector.Computer Engineering and Applications,2009,45(23):173-175.**

**Abstract:** For image-based rendering method toward people crowd rendering inside virtual scene,since the sampled images are limited,when viewpoint changes continuously in rendering process,there would appear leap problem.This paper proposes an image based rendering algorithm that uses Motion Vector to solve the problem,which produces new images for adjacent viewpoints by moving each pixel's position from the original images,achieves smooth transition between the discrete sampled images,solves the leap problem resulted from the limited sampled images,and increases rendering speed at the same time.Experiment shows that this method is valid.

**Key words:** virtual reality;people crowd simulation;image based rendering;Motion Vector

**摘要:** 针对虚拟场景中大规模人群图像绘制时由于采集图像数量有限,在视点连续变化时出现跳变的问题,提出了一种基于 Motion Vector 的图像绘制算法,该算法通过移动原始图像的像素位置生成相邻视点位置的图像,使得各个离散采样之间能够连续过渡,很好地解决了由于采样不足而引起的跳变问题,同时保证了绘制系统的实时性,实验结果验证了该算法的有效性。

**关键词:** 虚拟现实;人群仿真;基于图像的绘制;运动矢量

**DOI:**10.3778/j.issn.1002-8331.2009.23.048 **文章编号:**1002-8331(2009)23-0173-03 **文献标识码:**A **中图分类号:**TP391

## 1 引言

对于虚拟场景中大规模的人群绘制,为了提高实时性,系统选择图像建模与绘制的方法进行人群仿真。和传统几何建模与绘制方法相比,它极大地提高了绘制速度;但是在视点位置连续变化时,由于采样离散,出现了跳变现象,影响了绘制的真实感。

基于图像的绘制(Image-Based Rendering,IBR)技术,利用已有的场景图像生成新视点下的图像。与传统的几何绘制技术相比,它不需要进行复杂的几何建模,因而可使用较少的计算资源;它依赖的源图像可以是真实拍摄的图像,也可以是虚拟场景的截图,因此原始信息较容易获得;图像绘制的过程主要是图像处理 and 变换,因此绘制速度仅关于图像分辨率而不会

随着场景复杂度的增加而变化<sup>[1]</sup>。

提出了基于 Motion Vector 的图像绘制算法,并用于虚拟场景中大规模人群绘制。通过移动原始图像的像素位置生成相邻视点的图像,能够很好的解决由于采样不足而引起的跳变问题,并同时提高系统的实时性。

## 2 Motion Vector 算法

Yao Wang 在 Video Processing and Communications<sup>[2]</sup>中提出了运动模型概念。如图 1 所示, $C-XYZ$  为空间坐标系, $C$  为观察点, $xy$  为投影平面;假设模型上某点在三维空间的位置是  $X$ ,它在投影平面上的映射点为  $x$ 。当  $X$  移动到  $X'$  位置时, $X$  在投影平面上的映射点相应地从  $x$  移到了  $x'$ 。把 3D 位移  $D(X)=$

基金项目:北京市科委项目(No.D0605002040121);河北师范大学青年基金(No.L2007Q27)。

作者简介:霍奕(1982-),女,硕士,主要研究方向是虚拟现实与图形学。

收稿日期:2008-04-30 修回日期:2008-07-22

$X'-X=[Dx, Dy, Dz]$ , 叫做  $X$  点的 3D Motion Vector, 相应地把 2D 位移  $d(x)=x'-x=[dx, dy]$ , 叫做  $x$  点的 2D Motion Vector。

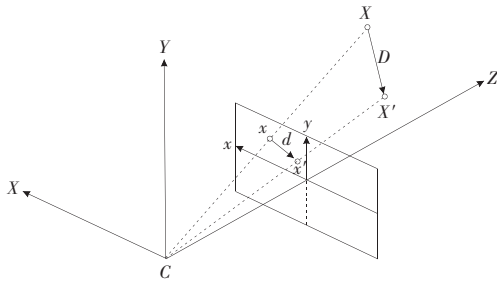


图1 投影对象的运动矢量

将 Motion Vector 思想应用于图像绘制; 计算出投影图像对应的 3D 模型经过空间位置移动后, 原图像上的每个投影点移动到的新位置, 对图像进行像素位置移动, 来生成新投影关系下的图像。图 2 为图像运动矢量示意图。第 2 章将根据摄像机透视投影模型结合 Motion Vector 的概念, 推导出基于 Motion Vector 的图像绘制公式和算法。

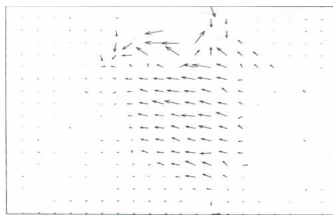


图2 2D 图像像素运动矢量

### 3 基于 Motion Vector 的图像绘制

在计算机上生成观察场景图像的变换过程可以和摄像机拍照的过程类比, 它们经历的变换过程如下:

- (1)观察变换: 摆放三脚架、放置照相机, 并且让照相机指向要拍摄的场景;
- (2)模型变换: 把要被拍摄的场景安排在合适的位置;
- (3)投影变换: 选择照相机的镜头或设置焦距;
- (4)视口变换: 决定最终照片的尺寸。

上述步骤完成后, 就可以用计算机绘制场景了(或拍摄照片)。下面几节将根据摄像机投影变换过程<sup>[2]</sup>推导基于 Motion Vector 的图像绘制算法。

#### 3.1 视口变换

最终显示在计算机屏幕的图像为  $M \times N$  数组。图 3 定义了一个屏幕坐标系  $u-v$ ,  $(u, v)$  是以像素为单位的屏幕坐标, 它表示了像素所在列、行, 而没有物理单位。以物理单位(如毫米)表示的投影平面坐标系为  $X-Y$ ; 该坐标系以屏幕范围内某一点  $O_1$  为原点,  $X$  轴、 $Y$  轴分别与  $u, v$  轴平行; 在  $X-Y$  坐标系中, 原点  $O_1$  定义为摄像机光轴与投影平面的交点, 该点一般位于图像中心处, 如图 3 所示。

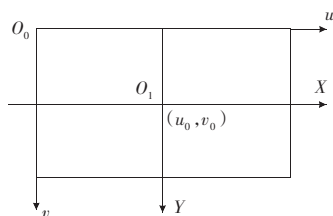


图3 屏幕坐标系与投影平面坐标系

从投影平面坐标系  $X-Y$  到屏幕坐标系  $u-v$  的变换过程相当于视口变换。假设每个像素在  $X$  轴与  $Y$  轴方向的物理尺寸分别为  $dX, dY$ , 则图像中的每个像素在屏幕坐标系与投影平面坐标系下的坐标有如下关系:

$$\begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = \begin{bmatrix} \frac{1}{dX} & 0 & u_0 \\ 0 & \frac{1}{dY} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} X \\ Y \\ 1 \end{Bmatrix} \quad (1)$$

### 3.2 投影变换

摄像机成像几何关系可由图 4 表示。其中  $O$  是摄像机光心,  $x$  轴和  $y$  轴与投影平面的  $X$  轴、 $Y$  轴平行,  $z$  轴为摄像机的光轴, 它与投影平面垂直。光轴与投影平面的交点是投影平面坐标系的原点。由点  $O$  与  $x, y, z$  轴组成的坐标系叫做摄像机坐标系。  $OO_1$  为摄像机的焦距  $f$ 。

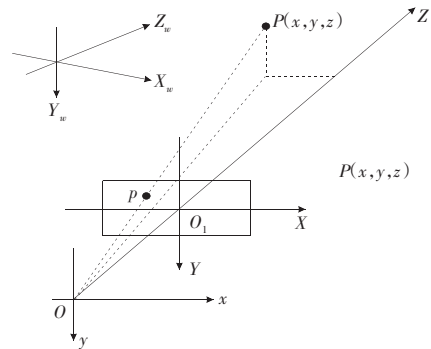


图4 摄像机坐标系与世界坐标系

计算机生成观察场景图像采用的是针孔相机模型。如图 4 所示, 假设  $P$  为摄像机坐标系中任意一点,  $OP$  与投影平面  $XY$  的交点为  $p$ ;  $p$  即为  $P$  的映射点。在这种关系下,  $P(x, y, z)$  与投影点  $p(X, Y)$  之间的关系为:

$$\begin{Bmatrix} X \\ Y \\ 1 \end{Bmatrix} = \begin{bmatrix} \frac{f}{z} & 0 & 0 & 0 \\ 0 & \frac{f}{z} & 0 & 0 \\ 0 & 0 & \frac{1}{z} & 0 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \quad (2)$$

由式(1)、(2)可得式(3), 即把摄像机坐标系中的一点  $P(x, y, z)$  代入式(3)可以求出它在屏幕坐标系下的坐标  $(u, v)$ 。其中  $P(x, y, z)$  为透视投影坐标系下的一点,  $M_2$  为透视投影变换矩阵,  $M_1$  为视口变换矩阵。

$$\begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = \begin{bmatrix} \frac{1}{dX} & 0 & u_0 \\ 0 & \frac{1}{dY} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{f}{z} & 0 & 0 & 0 \\ 0 & \frac{f}{z} & 0 & 0 \\ 0 & 0 & \frac{1}{z} & 0 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} = M_1 M_2 P \quad (3)$$

由式(3)的逆关系可以根据屏幕坐标系上的任意一点  $p(u, v)$ , 求出它所对应的 3D 模型点在摄像机坐标系中的坐标  $P(x, y, z)$ 。关系式如式(4)所示。

$$\begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} = \begin{bmatrix} \frac{z}{f} & 0 & 0 & 0 \\ 0 & \frac{z}{f} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dX & 0 & 0 & -u_0 dX \\ 0 & dY & 0 & -v_0 dY \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u \\ v \\ z \\ 1 \end{Bmatrix} = M_2^{-1} M_1^{-1} p \quad (4)$$

### 3.3 Motion Vector 算法流程

Motion Vector 算法的思想是利用图像的像素移动来模拟原三维模型的运动,算法流程(图 5 所示)如下。

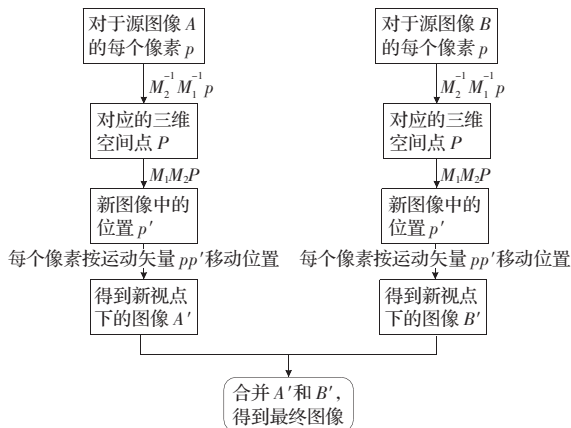


图 5 Motion Vector 算法流程图

采集相交的两个视线方向的图像  $A$ 、 $B$ ，分别对它们进行下列操作：

对于源图像中的每个像素  $p$ ，通过式(4)，根据它的屏幕坐标值  $(u, v)$  求出该像素点对应物体的三维坐标  $P(x, y, z)$ ；

根据物体运动模型对点  $P$  进行计算，求出它在三维模型移动后新的空间位置  $P'$ ；

对点  $P'$  利用公式(3)进行透视投影变换和视口变换，求出它在屏幕坐标系下的新位置  $p'$ ；

$pp'$  即为模型运动后某点在投影平面上的 2D Motion Vector；对源图像的每个像素进行上述 2D 运动计算，可以生成新投影关系下的图像  $A'$ 、 $B'$ ；

对  $A'$ 、 $B'$  进行合并计算，得到最终结果图像。

## 4 实验与分析

### 4.1 实验概况与数据

实验环境：操作系统平台：Windows XP，系统开发环境：Visual Studio.net，图形编程接口：OpenGL。

Motion Vector 算法公式： $C=f(A, \varphi_1)+f(B, \varphi_2)$ ， $A$ 、 $B$ ：相交视线方向上的两幅采样图像， $f: M_1 \cdot M_2 \cdot M_2^{-1} \cdot M_1^{-1} \cdot (A(p))$ ， $\varphi_1, \varphi_2$ ：视点变化的角度。

实验数据：首先采用单幅图像进行 Motion Vector 图像变换，设定采样图像  $A$  视线方向为  $0^\circ$ ，视线方向变化角度  $\varphi_1$  依次为  $5^\circ$ 、 $7^\circ$ 、 $10^\circ$ 、 $15^\circ$ 、 $30^\circ$ 、 $45^\circ$ 。

### 4.2 实验结果

采用单幅图像计算后，得到 Motion Vector 计算的中间结果如图 6 所示。

图 6(a)~(f)为真实三维模型旋转的结果，图 6(a')~(f')为使用 Motion Vector 算法对单幅图像计算的中间结果。通过对比得出，Motion Vector 算法可以正确模拟视点变化。但是，使用单幅图像会由于信息不足而产生空洞，并且这种现象随着旋转角度的增加而严重。

对相交视线方向上的两幅采样图像  $A$ 、 $B$  分别向目标视线方向进行 Motion Vector 变换(如图 7 所示)，再对中间结果  $A'$ 、 $B'$  合并，得到最终结果  $C$ 。实验发现，用于计算的两幅图像的采

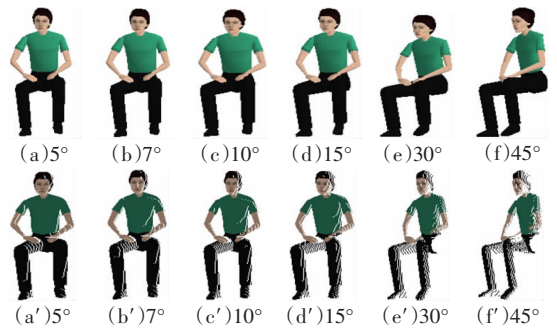


图 6 中间结果：使用单幅图像计算对比图

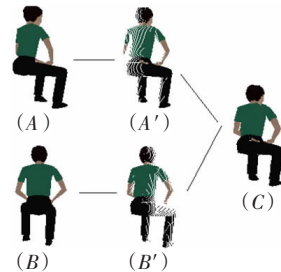


图 7 最终结果：两幅图像信息互补图

集视线方向夹角越小，绘制效果越好。

将 Motion Vector 算法用于虚拟场景中的大规模人群绘制，获得了很好的真实感，解决了单纯使用布告板图像绘制时出现的跳变问题；在视点漫游时，获得了连续变化的绘制效果。

### 4.3 实时性

对 Motion Vector 算法的实时性也进行了实验，并与布告板图像绘制、纯几何模型绘制的速度进行了比较，帧率比较结果如表 1 所示。对比发现，Motion Vector 算法在绘制速度上介于布告板图像绘制与纯几何方法之间，在真实感上可以达到几何绘制几乎相同的效果。

表 1 几种方法绘制速度比较表

绘制方法	人数				
	5	20	140	300	1000
基于视点的布告板	20	19	18.5	18	17
带深度的 MotionVector	15	14	12.5	12	11.5
纯几何模型	12	10.6	9	7.5	5

## 5 结论

对复杂虚拟场景中大规模的人群绘制，采用传统的图像绘制方法，相比几何绘制，可以极大的提高绘制速度；但会由于采样离散，在视点连续变化时出现跳变问题。提出的 Motion Vector 图像绘制算法，通过对采样图像的像素位置进行移动，可以得到任意视点的图像，从而得到了各个离散采样之间的连续过渡图像，有效解决了传统图像绘制时遇到的跳变问题。作为基于图像的绘制算法，Motion Vector 算法具有计算模型简单、参数易于获取，算法容易实现的特点。实验结果验证了该算法的有效性，能够有效解决图像绘制时由于采样不足而出现的跳变问题，并同时可以提高绘制系统的实时性。

### 参考文献：

- [1] 殷润民,李伯虎,柴旭东.虚拟现实中的基于图像绘制技术综述[J].系统仿真学报,2007,19(19).