

# 基于等密度映射和纹理重构的分层图像变换

汤 杨 吴慧中 肖 甫 肖 亮

(南京理工大学计算机科学与技术系, 南京 210094)

**摘 要** 图像变换包括图像映射和图像重构两个重要部分,而现有重构方法存在计算量较大、容易产生视觉误差等问题,并不能真正快速有效地实现图像变换。本文通过建立等密度映射的方式将重构转化为一个可以依赖纹理硬件快速完成的纹理映射过程,从而改进了传统的图像变换模型,并在此模型基础上设计了一种分层图像变换的方法。分层图像变换,首先将参考像素以相近的分辨率映射到对应的密度层以避免形成大量空洞,然后通过背景层设置、密度层合并等操作将对应的目的像素存储在一个可控长度的双向链表上,最后从后向前遍历链表逐层生成和合成纹理得到目的图像。实验结果表明,分层图像变换在有效避免视觉误差的基础上,大大提高了计算效率。

**关键词** 虚拟现实 基于图像的绘制 图像变换 等密度映射 纹理重构 分层

**中图法分类号:** TP391 **文献标识码:** A **文章编号:** 1006-8961(2006)04-0555-08

## Layered Image Warping Based on Equal-density Mapping and Texture-reconstruction

TANG Yang, WU Hui-zhong, XIAO Fu, XIAO Liang

(Department of Computer Science & Engineering, Nanjing University of Science & Technology, Nanjing 210094)

**Abstract** Image mapping and image reconstruction are two major components in performing Image warping. Existing reconstruction methods either involve huge computations or tend to produce visual errors in objects appearance using traditional warping model. This article presents a new image warping model by employing a novel "equal-density mapping" and "texture-Reconstruction". With the new model, a layered image mapping (LIW) approach is proposed then. In LIW, reference pixels are firstly mapped to "equal density layers" that parallel the desired view and are stored in a bidirectional link list after some by-layer processing. By implementing post-texture rendering along the list eventually, the warping is transformed to a fast texture mapping process, which provides a considerable speedup. Also, LIW is proved to efficiently tackle the "rubber sheet" problem without any pre-process.

**Keywords** virtual-reality, image-based rendering, image warping, equal-density mapping, texture-reconstruction, layered

## 1 引 言

图像变换是利用具有深度信息的采样参考图像来生成新视点下目的图像的过程。它是基于图像绘制 IBR (image-based rendering) 领域的研究热点,同时也是许多 IBR 系统<sup>[1-5]</sup>所依赖的基本原理。完整的图像变换包含图像映射和图像重构两个阶段,前

者是指根据几何约束建立参考和目的像素的对应关系并传递像素值,而后者是指通过对已有目的像素值的插值等操作构建出完整的目的图像。在图像变换中,第1步映射得到的目的图像常常会存在大量空洞,因此如何填补空洞更快更好地重构出目的图像就显得尤为重要。

空洞的产生有两个原因:一是由视点所引起的景深的改变使物体在目的视图中具有更高的分辨

基金项目:高等学校博士学科点专项科研基金项目(20020288024)

收稿日期:2005-01-14;改回日期:2005-06-02

第一作者简介:汤杨(1980~),男,现为南京理工大学计算机系博士研究生。主要研究方向为基于图像的绘制。E-mail: tang.yang@

率;二是由于遮挡等原因使得目的视图中的部分信息没有被参考图像所采样到。由于第 2 类空洞问题无法通过算法解决而必须要依赖其他更多的图像来提供被遮挡部分的信息,因此这里主要考虑在有限信息量下的第 1 类空洞以及视角变化较小情况下的第 2 类空洞。根据策略不同,现有的重构方法可分为基于模板 (splat) 和基于网格 (mesh) 两类(图 1)。

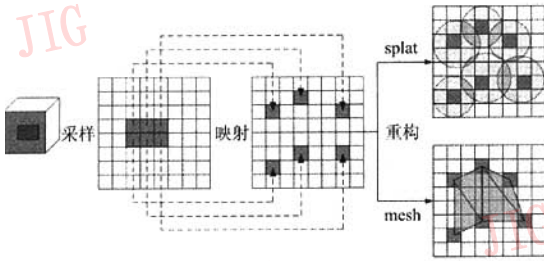


图 1 两种不同的重构策略

Fig. 1 Different strategies of image reconstruction

(1) 基于模板的算法 该类算法最早使用在体数据绘制中<sup>[6]</sup>,主要思想是根据参考像素的深度、方向等参数,对参考像素采用一个大小可变的内核进行重构。这样像素可以借助高斯内核<sup>[6]</sup>或其他内核<sup>[7,8]</sup>扩展为一块区域,再对重合的部分进行融合绘制,其中内核大小的计算可以针对单个像素<sup>[2]</sup>或场景节点的包围盒<sup>[7]</sup>进行估算。由于重构内核的多样性,基于模板的算法可以获得较好的绘制效果,但依旧存在不足:首先不区分空间结构信息,容易出现走样问题,而且,模板的计算完全依靠 CPU 进行,使得在高分辨率或复杂内核情况下计算量较大;另外,重构内核的尺寸不易处理,过大会降低显示效果且增加边缘融合的计算量,而过小则会在边缘处留下过多的细洞。

(2) 基于网格的算法 这类算法根据假设的空间连续性将离散的像素编制为一个连续的网格面,以像素作为网格的顶点,以插值计算的结果来填充网格面。该类算法可以借助硬件完成,但却无法有效地分辨空间的拓扑关系。当相邻像素并不位于同一空间连续表面时,就会产生不存在的空间表面,从而在场景的前景和后景之间常造成“橡皮擦现象”(rubber sheet)<sup>[1,10]</sup>。同时,该类算法总是需要构造一个与图像分辨率相当的面片数的场景,而不考虑这些面片本来的共面性,产生不必要的划分,从而增加了计算复杂度。

考虑到重构的过程和纹理映射存在相似性,改进传统的图像变换模型,其主要思想为(1)在映射阶段,参考像素不直接映射到目的视图上,而是保持原有分辨率不变地映射到目的域中相对应的成像面上,并保存为纹理信息,将这一映射方法称为等密度映射;(2)在重构阶段,根据目的视图的参数,直接对像素纹理实施纹理合成得到目的视图,这样的重构方式称为纹理重构。基于新模型,设计了一种分层图像变换的方法。和传统图像变换方法相比,在图像变换过程中,该方法仅让 CPU 完成了最必要的映射任务,而将大量的重构工作交由纹理硬件完成,从而大大提高了效率;同时,采用新模型实施图像映射时,分辨率和景深之间存在的联系自然地区分了目的像素间的连通性,从而无需相关预处理就有效地避免了“橡皮擦现象”。

## 2 等密度映射

采用传统映射方式,由于像素的采样密度不同,相同空间区域在目的域和参考域常常具有不同的分辨率(图 2)。而在变换中实际能达到的最高采样分辨率是由参考域决定的,因此,考虑到信息传递的充分性和必要性,将映射调整为只根据参考像素的采样密度进行,并称为等密度映射。

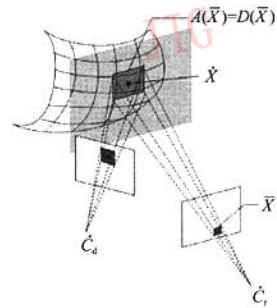


图 2 采样密度

Fig. 2 Sampling density

由于空间表面形状具有任意性,为了简化计算,采用像素  $\bar{X}$  在过空间采样点  $\dot{X}$  且与成像面平行的平面上的投影面积来表示其采样密度  $D(\bar{X})$ ,如图 2 所示,  $D(\bar{X}) = A(\bar{X})$ 。假设像素为单位矩形,则  $D(\bar{X})$  可表示为

$$D(\bar{X}) = 1 / \delta^2(\bar{X}) \quad (1)$$

其中,  $\delta(\bar{X})$  表示像素  $\bar{X}$  的普遍视差<sup>[10]</sup>。同时,在一

次映射  $f: \bar{X}_r \rightarrow \bar{X}_d$  中, 为了表示采样密度变化的程度, 引入映射密度比  $\rho$ , 定义为目的像素  $\bar{X}_d$  和对应参考像素  $\bar{X}_r$  的采样密度的比值的开方, 则有:

$$\rho = \sqrt{D(\bar{X}_d)/D(\bar{X}_r)}$$

由式(1)知

$$\rho = \delta(\bar{X}_r) / \delta(\bar{X}_d) \quad (2)$$

由此, 给出等密度映射的概念: 若映射  $f: \bar{X}_r \rightarrow \bar{X}_d$  为等密度映射, 则满足条件  $\rho = 1$ 。

由定义可知:

$$\delta(\bar{X}'_d) = \delta(\bar{X}_r) \quad (3)$$

### 3 基于等密度映射的图像变换模型

透射投影中, 成像面和视点构成一个视棱锥, 其在图像变换模型中被表示为相机矩阵  $P = [a, b, c]$ , 其中,  $a$  和  $b$  表示成像面中的坐标单位矢量,  $c$  表示从视点  $\dot{C}$  指向成像面坐标原点的矢量。在传统图像变换模型中, 参考像素被直接映射到目的相机矩阵的成像面上, 并在目的成像面分辨率下进行重构。如图3所示。

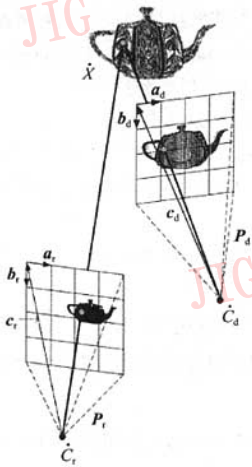


图3 传统图像变换模型  
Fig. 3 Traditional model

根据几何约束, 映射关系<sup>[10]</sup>建立为

$$P_d \bar{X}_d / \delta(\bar{X}_d) = (\dot{X}_r - \dot{X}_d) + P_r \bar{X}_r / \delta(\bar{X}_r) \quad (4)$$

新的模型采用等密度映射替代了直接映射的方式, 如图4所示, 参考像素  $\bar{X}_r$  首先被映射到目的视棱锥内某个中间层面上, 得到对应像素  $\bar{X}'_d$ 。

由于  $\bar{X}'_d$  和  $\bar{X}_r$  具有相近的采样密度, 采样信息

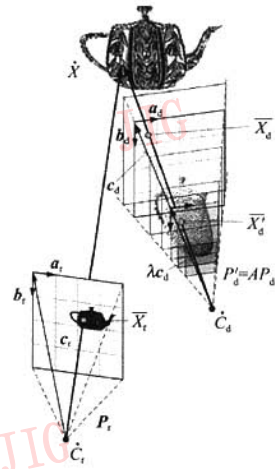


图4 等密度映射的图像变换模型

Fig. 4 Novel model

可以得到最有效和准确地传递, 同时, 若采用误差控制将相邻的像素映射到同一层面, 则可较好地保持参考像素间的连续性, 避免映射中大量空洞的出现。而由于中间层面和目的成像面具有相同的视点, 因此采用一个新的重构思路, 即将同一层面上的像素当成纹理, 直接映射到目的成像面上, 这样, 该过程即可保证原有的像素间的连续性, 又可以依赖纹理硬件快速完成。为了简化映射和重构过程, 中间层面设定为视棱锥内平行于目的成像面的平面, 通过引入一个调节矩阵  $A$  建立其和目的相机矩阵间的关系为

$$P'_d = AP_d$$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \lambda \end{bmatrix} \quad (\lambda > 0, \text{称为密度调节系数})$$

不难发现  $\lambda$  的作用在于动态改变映射的  $c_d$  分量, 而保证  $a_d$  和  $b_d$  分量不变, 从而使得  $\bar{X}'_d$  在不同的  $P'_d$  中具有不同的普遍视差  $\delta(\bar{X}'_d)$ 。这样对于某个给定的  $\delta(\bar{X}_r)$ , 总能找到适当  $\delta(\bar{X}'_d)$  使得等密度映射(式(3))得到满足。在新模型中讨论以下几个问题:

(1) 映射方程的建立

由于符合相同的几何约束, 由式(4)知映射关系满足

$$P'_d \bar{X}'_d / \delta(\bar{X}'_d) = (\dot{X}_r - \dot{X}_d) + P_r \bar{X}_r / \delta(\bar{X}_r)$$

由式(3)该关系可表示为

$$P'_d \bar{X}'_d = \delta(\bar{X}_r) (\dot{X}_r - \dot{X}_d) + P_r \bar{X}_r$$

(2) 密度调节系数  $\lambda$

不失一般性, 图5给出了新模型中极平面内的

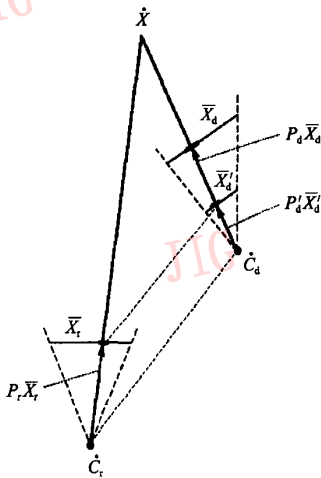


图 5 密度调节系数的计算

Fig. 5 Calculation of coefficient

映射关系。若  $\bar{X}_i \rightarrow \bar{X}'_d$  为等密度映射,由相似性原理可得  $\lambda = \delta(\bar{X}'_d)/\delta(\bar{X}_i)$ 。又由式(3)知  $\delta(\bar{X}'_d) \approx \delta(\bar{X}_i)$ , 所以

$$\lambda = \delta(\bar{X}_i)/\delta(\bar{X}_d) = \rho \quad (5)$$

在实际处理时,由于图像变换时视角的变化不能过大,同时参考和目的相机矩阵常具有相同的内部参数,所以直接使用空间点与视点的距离来计算  $\lambda$ , 即

$$\lambda = \delta(\bar{X}_i)/\delta(\bar{X}_d) = |\overline{C_d X_i}|/|\overline{C_i X_d}| \quad (6)$$

这样,具有相同映射密度比  $\rho$  的参考像素会映射到相同的成像面上,因此也称该成像面为密度层  $Layer(\lambda)$ 。

#### (3) 不同密度层之间的关系

若参考像素  $\bar{X}_i$  映射到不同密度层  $Layer(\lambda^{(m)})$  和  $Layer(\lambda^{(n)})$ , 得到对应目的成像平面的像素坐标为  $\bar{X}_d^{(m)}(u_d^{(m)}, v_d^{(m)})$  和  $\bar{X}_d^{(n)}(u_d^{(n)}, v_d^{(n)})$ , 依据相似性原理不难推出:

$$u_d^{(m)}/u_d^{(n)} = \lambda^{(m)}/\lambda^{(n)}, v_d^{(m)}/v_d^{(n)} = \lambda^{(m)}/\lambda^{(n)} \quad (7)$$

其中,  $m, n$  表示不同密度层的层号。

#### (4) 纹理重构坐标

对位于密度层  $Layer(\lambda)$  的像素纹理  $\bar{X}'_d(u'_d, v'_d)$  进行重构,可得到目的图像平面的一块矩形区域。由式(7)可知被映射矩形区域位于目的图像空间相应的左上和右下坐标分别为  $(u'_d/\lambda, v'_d/\lambda)$  和  $((u'_d + 1)/\lambda, (v'_d + 1)/\lambda)$ 。

(5) 由密度调节数的定义知,若  $\lambda = 1$ , 参考像素直接映射到目的图像平面,回归到传统的映射方式。

## 4 分层的图像变换

采用新的图像变换模型,分层图像变换过程如下:在映射阶段中,首先设定目的图像平面( $\lambda = 1$ )为背景层,依据 list-priority 算法<sup>[12]</sup>确定的映射顺序,将参考像素分别映射到背景层或密度层  $Layer(\rho)$  上,然后通过层次生成和选择算法将像素存储在一个可控长度的密度层双向链表上;在重构阶段,首先利用不同的连通性在层内和层间分别处理空洞,然后逆向合成各层,得到目的图像。

### 4.1 图像映射

#### 4.1.1 背景层的设置

若完全按照等密度映射,当  $\rho > 1$  时,参考像素会映射到比目的图像分辨率还高的密度层上,显得相对冗余,因此在分层算法中,首先确定  $\lambda = 1$  的层为背景层。当  $\rho > 1$  就映射到背景层上,否则由式(5)映射到内部  $\lambda = \rho$  的等密度层。

#### 4.1.2 密度层结构

各密度层的结构包含如下信息:

```
class CLayer {
float den_x; //该密度层对应的映射密度比
unsigned char * data; //指向该层数据存储区的指针
int layer_height; //层高分辨率
int layer_width; //层宽分辨率
int next_layer; //静态链表的指针,指向后一个密度层
int pre_layer; //静态链表的指针,指向前一个密度层
CLayer();
CLayer(int wid,int hei,float den);
virtual ~CLayer();};
```

其中,由式(5)可知  $den\_x$  即该层的密度调节系数  $\lambda$ , 由式(7)知在密度层和目的图像的分辨率存在关系:

$$layer\_height = image\_height * den\_x,$$

$$layer\_width = image\_width * den\_x.$$

这样各层  $data$  指向的空间大小也相应为  $sizeof(pixel) * layer\_height * layer\_width$ 。本文为了显示需要,设置了  $4 * unsigned\ char$  的 RGBA 像素。

#### 4.1.3 密度层生成与选择

如图4,若完全按照参考像素的映射密度比来产生对应的密度层,显然过多的层数会大大降低计算和存储的开销,因此,本文通过一个层次精度系数在误差允许范围生成和选择各密度层。

如图6所示,若参考像素  $\bar{X}_i$  映射到相应密度层

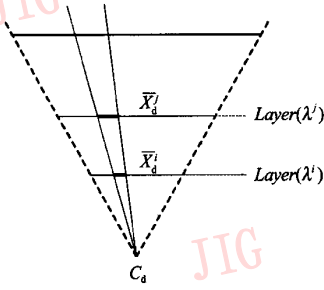


图 6 密度层的精度调节  
Fig.6 Adjust of layers

Layer( $\lambda^i$ ) 得到对应像素  $\bar{X}_d^i$ , 为了保证层次调节时不会出现新的空洞, 设密度层 Layer( $\lambda^j$ ) 上的对应像素为  $\bar{X}_d^j$ , 则对应像素区域大小应满足:  $|\bar{X}_d^j| - |\bar{X}_d^i| < \xi$  ( $\xi = 1$ , 称  $\xi$  为层次精度系数, 像素设为单位矩形)。而在实际处理时, 考虑到周围像素的相互作用, 一般  $\xi \leq 0.5$ 。又由式(7)可知,  $|\bar{X}_d^j| / |\bar{X}_d^i| = \lambda^j / \lambda^i$ , 则可推得像素  $\bar{X}_d^j$  的可调节密度层范围满足  $\lambda^j \in [(1 - \xi/2)\lambda^i, (1 + \xi/2)\lambda^i]$ 。这样, 当在像素的可调节范围内已存在密度层, 则直接映射到该密度层, 否则就新建一个密度层。采用静态双向链表记录各密度层, 该步算法相应的伪语言描述如下:

```
int GetLayer( $\rho, \xi$ )
{
    if( $\rho > 1$ ) return 0; //当密度比大于 1 返回背景层 Layer[0]
    int i = first_layer; //i 设置为第一层
    while(Layer[i] existed) {
        if( $\rho > \text{Layer}[i].\text{den}_x$ ) {
            i = Layer[i].next_layer; // i 指向下一层
            continue; //若  $\rho$  比当前层的密度比大则继续
        }
        if(Layer[i].den_x -  $\rho < \xi/2$ 
        or  $\rho - \text{Layer}[\text{Layer}[i].\text{pre\_layer}].\text{den}_x > \xi/2$ )
            return i or Layer[i].pre_layer;
        //合并范围内寻找合并层并返回
    }
    else {
        Layer[curr_layer_max] = new CLayer();
        //超出合并范围, 在已有层数 curr_layer_max 上创建新层
        Layer[Layer[i].pre_layer].next_layer = curr_layer_max;
        Layer[curr_layer_max].pre_layer = Layer[i].pre_layer;
        Layer[i].pre_layer = curr_layer_max;
        Layer[curr_layer_max].next_layer = i;
        //调整指针将新建层插入双向链表
        return curr_layer_max ++ ;
    }
}
```

```
//返回新建层, 总层数加 1
}
}return ERROR;
}
```

这样, 不同的层次精度系数  $\xi$  使得在映射阶段会形成不同数量的密度层, 同时映射使得各层按照映射密度比的大小进行升序排列, 形成了一个以背景层为尾节点的静态链表。

### 4.2 图像重构

在重构阶段主要完成两个任务: 第一, 在层内和层间分别填补细洞; 第二, 将各密度层合成到背景层上, 得到目的视图。

#### 4.2.1 空洞的填补

由于等密度映射在映射过程几乎没有进行分辨率的扩展, 因而层内的第 1 类空洞已几乎没有。但由于采用了层的合并, 当精度设置过低时会造成像素映射到分辨率差异较大的密度层上, 造成部分第 1 类空洞。由于该类空洞的数量非常有限, 因此就直接采用线性插值的方法进行填补。而对于第 2 类空洞, 由于图像变换的原理限制了视角的变化不大, 因此采用和文献[1]类似的方法来生长不同密度层的边缘像素来进行填补, 原理如图 7 所示。

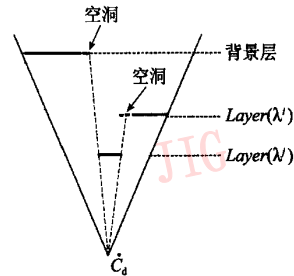


图 7 层间空洞的填补  
Fig.7 Filling holes between layers

#### 4.2.2 逆向的密度层合成

利用纹理硬件可以简单而快速的将绘制好的各个密度层合成出目的图像, 但是如何保证正确的可见性是合成中需要考虑的问题。

对于同一密度层内像素之间的遮挡关系, 由于采用 list-priority 算法原理进行排序, 可保证在相同目的视线上, 像素从远到近进行映射, 具体证明可见文献[10]。

而对于各密度层之间的遮挡关系, 研究发现, 在视角变化不大时, 只要按照静态链表的顺序反向从链表尾到链表头逐层绘制, 就可保证在不使用 z-buffer

的情况下正确地处理遮挡关系。下面给出简单证明:

如图 8 所示,参考像素  $\bar{X}_r^1$  和  $\bar{X}_r^2$  分别对应空间点  $\dot{X}^1$  和  $\dot{X}^2$ ,其在目的视点  $C_d$  观察方向上形成前后遮挡,则要证明调节系数小的密度层对应的空间点离目的视点更近,即:

若  $0 < \rho(\bar{X}_r^1) < \rho(\bar{X}_r^2) < 1$ , 则有  $|\overrightarrow{C_d X^1}| < |\overrightarrow{C_d X^2}|$ 。

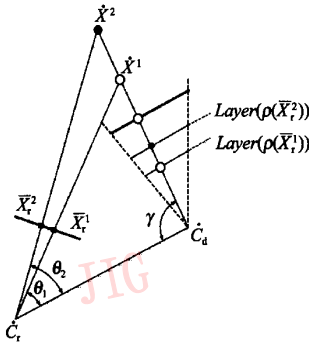


图 8 可见性说明

Fig.8 Visibility explanation

**证明** 由式(6)知  $\rho(\bar{X}_r^1) = |\overrightarrow{C_d X^1}| / |\overrightarrow{C_r X^1}| = \sin(\theta_1) / \sin(\gamma)$ , 同理  $\rho(\bar{X}_r^2) = \sin(\theta_2) / \sin(\gamma)$ 。

由于  $\rho(\bar{X}_r^1) < \rho(\bar{X}_r^2)$ , 所以  $\sin(\theta_1) < \sin(\theta_2)$ 。又由于  $0 < \rho(\bar{X}_r^1) < 1$  且  $\theta_1 + \gamma < \pi$ , 所以  $0 < \theta_1 < \pi/2$ , 同理  $0 < \theta_2 < \pi/2$ , 有  $0 < \theta_1 < \theta_2 < \pi/2$ 。

由几何关系知

$$|\overrightarrow{C_d X^1}| / |\overrightarrow{C_d C_r}| = \sin(\theta_1) / \sin(\pi - \theta_1 - \gamma)$$

$$|\overrightarrow{C_d X^2}| / |\overrightarrow{C_d C_r}| = \sin(\theta_2) / \sin(\pi - \theta_2 - \gamma)$$

两式相比得

$$\frac{|\overrightarrow{C_d X^1}|}{|\overrightarrow{C_d X^2}|} = \frac{\sin(\theta_1) \sin(\theta_2 + \gamma)}{\sin(\theta_2) \sin(\theta_1 + \gamma)}$$

$$= \frac{\sin(\theta_1) (\sin(\theta_2) \cos(\gamma) + \cos(\theta_2) \sin(\gamma))}{\sin(\theta_2) (\sin(\theta_1) \cos(\gamma) + \cos(\theta_1) \sin(\gamma))}$$

上下同除  $\sin(\theta_1) \sin(\theta_2)$  有

$$\frac{|\overrightarrow{C_d X^1}|}{|\overrightarrow{C_d X^2}|} = \frac{\cos(\gamma) + \cot(\theta_2) \sin(\gamma)}{\cos(\gamma) + \cot(\theta_1) \sin(\gamma)}$$

因为  $0 < \theta_1 < \theta_2 < \pi/2$ , 所以  $0 < \cot \theta_2 < \cot \theta_1$ ,  $|\overrightarrow{C_d X^1}| / |\overrightarrow{C_d X^2}| < 1$ , 即  $|\overrightarrow{C_d X^1}| < |\overrightarrow{C_d X^2}|$ 。

## 5 实验与分析

在 P4 1.7 的 PC 上实现了分层的图像变换算法,同时实现了传统的变换算法进行比较。实验的目的在于从效果和效率上检测本文方法的有效性。

采用 OpenGL 构造了一个茶壶场景,在不同深度上放置了 3 个茶壶。图 9(a)为初始的采样参考图像,经过第 1 步映射得到图 9(b),显然图 9(b)中存在大量空洞。图 9(c)、(d)、(e)表示了采用不同方式进行重构的结果。其中,图 9(c)是采用 Gauss 内核的模板重构方法,在内核边缘的融合部分留下了一些细纹,采用更复杂的内核可以获得更好的效果;图 9(d)是网格重构的结果,由于茶壶边缘像素和背景像素深度相差较大,所以“橡皮擦”现象很明显;图 9(e)是本文方法采用纹理重构的结果,其中采用了 OpenGL 的线性滤波器,  $\xi = 0.2$ , 变换生成的密度层如图 9(f)所示。

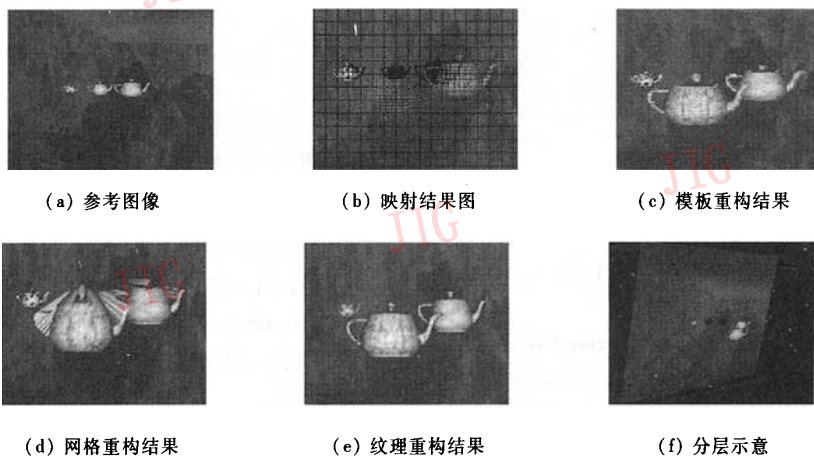


图 9 采用不同方法进行变换

Fig.9 Results of image warping

对茶壶场景的各种方法的效率进行了统计(表 1)。其中 LIW(layered image warping)表示本文采用分层图像变换方法,SP(splat)表示采用 Gauss 内核的 splat 方法,ME(mesh)表示采用 mesh 重构的方法,由于具有相同的原理,直接采用了 OpenGL 中的像素重投影过程进行映射。

表 1 计算效率统计

Tab.1 Performance statistics unit 单位:ms

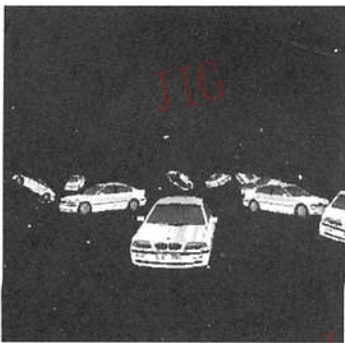
	方法	映射耗时	重构耗时	总耗时
Test1 (512 × 512)	LIW	644	127	771
	SP	571	992	1 563
	ME	571	1 172	1 743
Test2 (320 × 320)	LIW	265	96	361
	SP	237	360	597
	ME	237	432	669
Test3 (256 × 256)	LIW	183	32	215
	SP	162	189	351
	ME	162	248	410

由表 1 可见,本文算法在映射阶段虽然由于需要确定密度层而增加了部分开销,但在重构阶段只需完成纹理数据的装载和纹理映射,而无需使用 CPU 逐像素进行重构,因此大大缩短了重构时间,从而使得整个过程的计算效率提高 50% 左右。

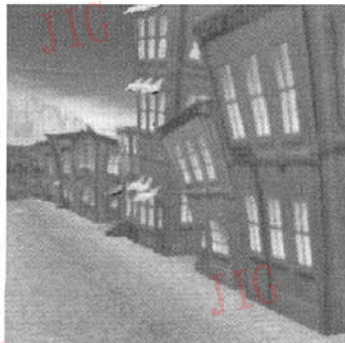
增加场景的复杂度,对 BMW 场景和 Street 场景进行了进一步测试,最终结果和效率统计如图 10 所示,其中, BMW 场景的  $\xi = 0.2$ ,纹理滤波方式采用 GL\_LINEAR; Street 场景  $\xi = 0.25$ ,纹理滤波方式采用 GL\_NEAREST。

实验还对各个场景设定不同的  $\xi$  进行测试,测试的结果表明当层数不多时对效率的影响不是很明显,而在层数超过 10 层或更多时,加速效果会明显下降。研究发现每次变换的最佳  $\xi$  值并不稳定,而是和各个场景特征相关联的,在一般情况下取  $0.1 \leq \xi \leq 0.3$ ,层数控制在 5 层左右就能获得不错的效果。

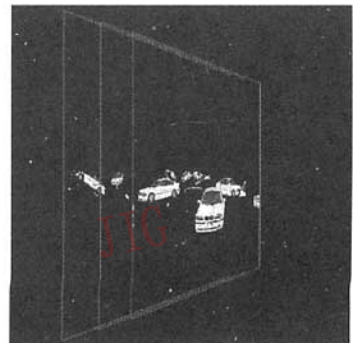
在成像效果方面,由于重构的过程是建立在等密度映射的基础上,而等密度映射是从参考采样密



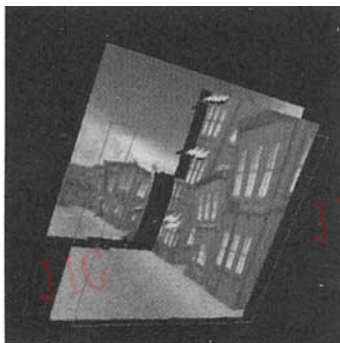
(a) BMW 场景的变换结果



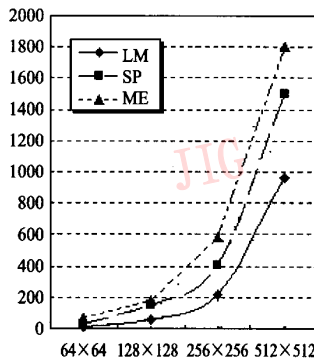
(b) Street 场景的变换结果



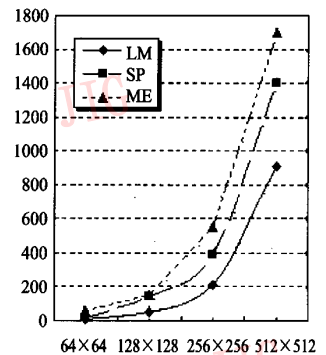
(c) BMW 的变换分层示意



(d) Street 的变换分层示意



(e) BMW 的效率统计



(f) Street 的效率统计

图 10 其他场景的测试

Fig.10 Test with other scenes

度出发的,已经保证了采样信息的有效传递,因此采用纹理映射的方式进行重构不会影响有效信息从参考域向目的域的传递,从而不会降低目的图像的成像质量。

## 6 结 论

利用重构和纹理变换的相似性,采用新的图像变换模型,提出了一种分层的图像变换方法。和分层的图像缓存<sup>[13,14]</sup>或分层 imposter<sup>[15,16]</sup>方法不同的是,原有方法在参考域内,通过对参考像素深度的预处理,预先形成多个深度层次,然后再利用同一深度层次的共性进行加速;而本文方法则不需预处理,在目的域中形成多个密度层。这样,在需要进一步变换时,可以将本文映射的中间结果保留在各个密度层上,直接作为图像缓存或分层 imposter 方法的输入。同时,采用纹理变换的重构方式,除了可以提高效率并避免“橡皮擦现象”外,还可以直接通过对硬件控制方便快速地改变目的图像的效果。

在本文方法的进一步讨论中,密度层不一定需要平行于目的成像面,而可以平行于参考成像面,这样更可以利用文献[17]的原理进一步提高 Image Warping 的计算效率;另外,如何能通过对参考场景特征的预处理自动获得最佳的  $\xi$  从而获得最好的效果也是下一步要研究的目标。同时,在今后的工作中还将进一步研究多参考图,多视点成像的加速算法以及更优的图像处理算法。

### 参考文献 (References)

- 1 Mark W R, McMillan L, Bishop G. Post-rendering 3D warping[A]. In: Proceedings of the 1997 Symposium on Interactive 3D Graphics [C], New York, NY, USA, 1997:7~16.
- 2 Shade J, Gortler S, He L, et al. Layered depth images[A]. In: Proceedings of the SIGGRAPH'98[C], Orlando, FL, USA, 1998: 231~242.
- 3 Rafferty M M, Aliaga D G, Lastra A A. 3D image warping in architectural walkthroughs [A]. In: Proceedings of the Virtual Reality Annual International Symposium [C], Atlanta, GA, USA, 1998:228~233.
- 4 Zhang Yan-ci, Liu Xue-hui, Wu En-hua. Accelerated backward warping[J]. Journal of Computer Science and Technology, 2003, 18(1):48~55.
- 5 Zhang Yan-ci, Liu Xue-hui, Wu En-hua. Plane-based warping[A]. In: Proceedings of Seventh International Conference on Computer Aided Design and Computer Graphics [C], Kunming, Yunnan, China, 2001: 257~266.
- 6 Lee Westover. Footprint evaluation for volume rendering [J]. Computer Graphics, 1990, 24(4):367~376.
- 7 Swan J, Mueller K, Möller T, et al. An anti-aliasing technique for splatting[A]. In: Proceedings of the 8th Conference on Visualization '97[C], Phoenix, Arizona, USA, 1997:197~205.
- 8 Pfister H, Zwicker M, Van Baar J, et al. Surfels: surface elements as rendering primitives [A]. In: Proceedings of SIGGRAPH '00 [C], New Orleans, Louisiana, USA, 2000:335~342.
- 9 Chang C F, Bishop G, Lastra A. LDI tree: a hierarchical representation for image-based rendering [A]. In: Proceedings of SIGGRAPH'99[C], New York, NY, USA, 1999:291~298.
- 10 McMillan L. An Image-based Approach to Three-dimensional Computer Graphics[D]. Chapel Hill, NC, USA: University of North Carolina, 1997.
- 11 McMillan L, Bishop G. Plenoptic modeling: an image-based rendering system[A]. In: Proceedings of the SIGGRAPH'95[C], Los Angeles, SA, USA, 1995:39~46.
- 12 McMillan L. Computing Visibility Without Depth. Technical [R]. TR95-047, Chapel Hill, NC, USA: University of North Carolina at Chapel Hill, Dept. of Computer Science, 1995.
- 13 Schaufler, Gernot, Wolfgang Stürzlinger. A three-dimensional image cache for virtual reality[A]. In: Proceedings of the 7th Eurographics Workshop on Rendering '96[C], Porto, Portugal, 1996: 227~236.
- 14 Max, Nelson. Hierarchical rendering of trees from precomputed multi-layer z-buffers [A]. In: Proceedings of the 7th Eurographics Workshop on Rendering '96[C], Porto, Portugal, 1996:165~174.
- 15 Jed Lengyel, John Snyder. Rendering with coherent layers[A]. In: Proceedings of the SIGGRAPH'97[C], Los Angeles, SA, USA, 1997: 233~242.
- 16 Chaufler G. Per-object image warping with layered impostors [A]. In: Proceedings of the 9th Eurographics Workshop on Rendering[C], Vienna, Austria, 1998:151~162.
- 17 Oliveira, Manuel M, Gary Bishop. Factoring 3-D Image Warping Equations into a Pre-Warp Followed by Conventional Texture Mapping [R]. TR99-002, Chapel Hill, NC, USA: University of North Carolina, 1999.