

用于图像逆映射的搜索转移算法^{*})

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 图像逆映射是基于图像绘制技术的基础算法,但由于需要进行繁琐的参考域搜索过程,传统图像逆映射算法计算复杂度大、效率低。本文提出一种快速图像逆映射算法——搜索转移算法。该算法从场景的极线几何性质出发,将大量匹配搜索过程从参考域转移到目的域中进行,由于在目的域只需进行简单的判断操作,因此可以大大降低整体的匹配复杂度;同时,算法充分利用场景的空间连贯性,通过为目的域制定自适应的绘制顺序,有效地减小了实际需要搜索的步长。实验表明,搜索转移算法简单有效,能极大地降低搜索复杂度 95% 以上,提高计算效率 10 倍左右。

关键词 虚拟现实,计算机视觉,基于图像的绘制,搜索转移

A Novel Search-Transform Algorithm for Inverse Image Warping

TANG Yang WU Hui-Zhong XIAO Fu XIAO Liang

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

Abstract As the basic algorithm of image-based rendering, inverse image warping is blocked to wide application for its high complexity and low performance, which is due to the complicate search-and-match process in reference field. In this paper, a novel search-transform algorithm (STA) is proposed to change the situation. On the observation that search-and-march process in desired field is much simpler than reference one, STA transform most of this process from reference field to desired one based on the epipolar geometry characters. Meanwhile, STA takes full advantage of the spatial continuity of scene to effectively reduce the step length of searching by setting the self-adaptive warping order for desired pixels. STA is proved to be spare but efficient by experiments, which can eliminate most of the redundant computation and obviously accelerate the traditional inverse image warping algorithm.

Keywords Virtual-reality, Compute vision, Image-based rendering, Search-transform

1 引言

随着人们对虚拟现实真实感和沉浸感要求的提高,数以千万甚至亿计的场景面片已使得图形设备不堪重负。基于图像的绘制技术 IBR (image-based rendering) 由于具有与场景复杂度无关和高度真实感等特点,已成为融合图像和图形领域的新的研究热点。图像映射 (image warping)^[4,5] 是众多 IBR 漫游系统^[1~3] 采用的基础算法,是根据带有深度信息的参考图像来生成新视点下的目的图像。提高图像映射算法的效率是优化 IBR 系统性能的根本动力。

图像映射可分为正向映射和逆向映射。正向映射从参考域出发,根据映射方程从参考像素直接计算出对应目的像素的位置,建立匹配映射,并将采样信息从参考域传递到目的域。正向映射在单幅参考图像时速度较快,但场景信息常由多幅参考图像提供,使得计算开销倍数增长。同时,由于映射中采样密度的不同,采用正向映射还存在空洞、多射及映射出界等问题^[1,2,4,5]。逆向映射从目的域出发,逆向求取匹配的参考像素,从不同的参考域进行采样信息的传递。由于采用了显示驱动的原理,逆向映射很好地解决了正向映射的诸多问题,但由于目的域的深度信息未知,无法直接建立匹配映射,而需要在参考域中进行搜索匹配,造成了逆向映射的效率瓶颈^[4,6~9]。

本文从场景的极线几何特性出发,提出一种快速图像逆映射算法——搜索转移算法 (Search-transform Algorithm)。该算法利用极线保序性和场景的空间连贯性,将传统算法在

参考域中进行的大量匹配搜索运算转移到目的域中进行。由于目的域搜索过程的复杂度大大低于参考域,而空间连贯性又显著降低参考域的搜索长度,因此本文算法通过降低参考域搜索复杂度极大地降低了整个算法的复杂度,显著提高了计算效率。

2 相关概念与性质

图像映射的本质是信息量在参考和目的域中转移的过程。不妨令参考域表示为 $R_r(C_r, P_r, Col_r, \delta_r)$, 目的域表示为 $R_d(C_d, P_d, Col_d, \delta_d)$, 其中 $C_i, P_i, Col_i, \delta_i$ ($i=r$ 或 d) 分别表示视点、相机矩阵、采样信息即普遍视差。这样,图像映射问题就转化为一个已知参考域的所有信息 $R_r(C_r, P_r, Col_r, \delta_r)$ 及目的域的视点 C_d 和相机矩阵 P_d , 来求取目的采样信息 Col_d 的过程。问题的关键是建立匹配参考和目的像素间的匹配映射关系 $f(X_r^i) = X_d^i$ 。依据空间的极线几何约束关系 (图 1), 正向和逆向映射方程^[4] 分别表示为:

$$\bar{X}_d^i = \delta(X_r^i) P_d^{-1} (C_d - C_r) + P_d^{-1} P_r X_r^i \quad (1)$$

$$\bar{X}_r^i = \delta(X_d^i) P_r^{-1} (C_r - C_d) + P_r^{-1} P_d X_d^i \quad (2)$$

其中,由于式(2)中 $\delta(X_d^i)$ 未知,因此逆向方程无法直接求解,需要在参考域中进行逐像素的搜索验证来进行匹配。而由极线几何关系可知 X_r^i 必然位于参考视点 C_r 、目的视点 C_d 和 X_d^i 所决定的平面 (称为极平面) 内,因此极平面和参考图像交线 (称为极线) 的约束将搜索从二维降到一维,即沿极线进行搜索。这种极线几何关系是图像映射中的基本约束条件,具备以下基本性质^[10,11]:

^{*} 本课题得到高等学校博士学科点专项科研基金(20020288024)资助。汤 杨 博士研究生,主要研究方向为基于图像的绘制;吴慧中 教授,博士生导师,主要研究方向为虚拟现实和系统仿真;肖 甫 博士研究生,主要研究方向为计算机图形图像理论;肖 亮 博士、讲师,主要研究方向为计算机图形图像理论。

(1) 整体匹配性 设映射中极线 l_r 和 l_d , 像素 X_r^i 和 X_d^i ($i=1, 2, \dots, n$) 相互对应, 若 $X_r^i \in l_r$, 则 $X_d^i \in l_d$ 。即参考图像中极线上对应点位于目标图像的极线上, 反之亦然。这样, 所有映射关系可以认为在相互独立的极平面进行。

(2) 极线保序性 空间点 X^i 投影到参考和目的图像平面分别得到像素 \bar{X}_r^i 和 \bar{X}_d^i ($i=1, 2, 3$), 则其在极线上排列顺序是一致的。需要说明的是, 由于漫游时视点变化的连续性, 使得极线保序性在大部分情况下严格成立, 而对于视差梯度相差很大或包含透明元素的场景不是严格成立。本文针对一般场景进行讨论。

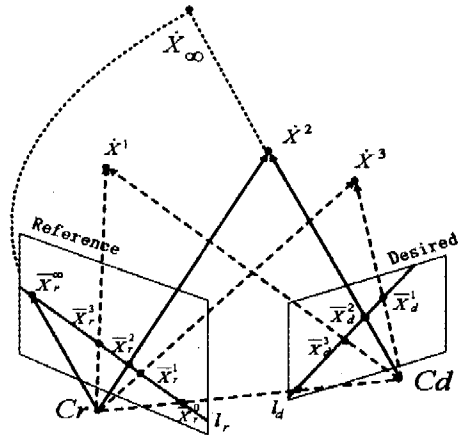


图1 极线几何性质

3 本文算法

3.1 算法原理

传统图像逆映射算法中, 对于每个目的像素 X_d^i , 需要沿着对应参考域的极线从 X_r^0 向 X_r^∞ 方向进行逐像素搜索验证 (如图 2a)。本文算法则利用极线几何性质, 先在目的域中沿极线逆向寻找已建立的匹配映射, 然后再转移到参考域进行搜索, 这种搜索域的转移过程可以避免大量的计算。其具体原理 (如图 2b) 表述为: 对于每个目的像素 X_d^i , 首先沿极线 l_d 逆向搜寻到已建立匹配映射的最近的目的像素 $X_d^{i-\sigma}$ (其中 $f^{i-\sigma}(X_d^{i-\sigma}) = X_d^i$, 称 $X_d^{i-\sigma}$ 为 X_d^i 的最近相邻像素, σ 为一个较小的正整数), 由极线整体匹配性可知 $X_d^{i-\sigma} \in l_r$ 。然后以 $X_d^{i-\sigma}$ 为起点, 搜索到目的像素 X_r^i , 从而建立映射 $f^i(X_r^i) = X_d^i$ 。最后依据所建立的映射传递采样信息 $Col_d(X_d^i) = Col_r(X_r^i)$ 。

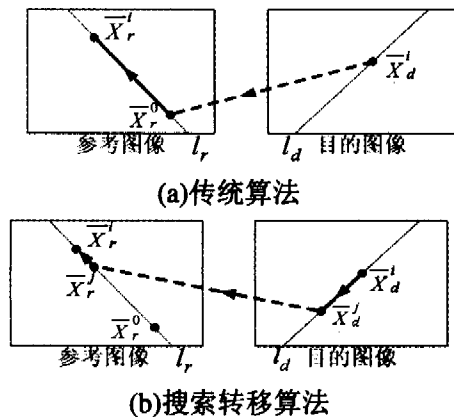


图2 算法原理

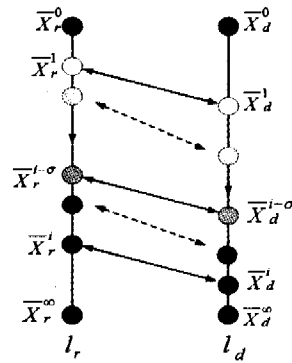


图3 信息量有效性

3.2 信息量的有效性

由于本文算法改变了参考极线搜索的起点, 因此如何保证有效的采样信息仍位于待搜索的极线段上, 是需要进行讨论的问题。图 3 表示了对应极线 l_d 和 l_r 上的匹配关系, 其中 $f^i(X_r^i) = X_d^i$ ($i=0, 1, 2, \dots, n$)。若 X_r^i 可见而 X_d^i 不可见, 则 $X_d^i = \phi$, 反之 $X_r^i = \phi$ 。由图像映射的空间几何约束知, 对于目的视点 C_d , l_r 存在从远到近的排列 $L_r(X_r) = \{X_r^0, X_r^1, \dots, X_r^{i-\sigma}, \dots, X_r^i, \dots, X_r^j, \dots, X_r^\infty\}$ 。同理, 目的极线 l_d 的排列为 $L_d(X_d) = \{X_d^0, X_d^1, \dots, X_d^{i-\sigma}, \dots, X_d^i, \dots, X_d^j, \dots, X_d^\infty\}$ 。当本文算法沿着 l_d 逆序搜索时 (这里将 i 增加的方向记为顺序, 反之为逆序), 有 $X_d^i \in \overline{X_d^{i-\sigma} X_d^\infty}$, 即 $0 \leq i-\sigma < i \leq \infty$ 。由极线几何的保序性可知 $X_r^i \in \overline{X_r^{i-\sigma} X_r^\infty}$, 即待搜索到的目的像素必然位于以 $X_r^{i-\sigma}$ 为起点、 X_r^∞ 为终点的极线段上, 或者不存在采样信息 $X_r^i = \phi$ 。

3.3 目的域绘制顺序

本文算法在确定匹配关系时首先将搜索转移到了目的域。这里, 我们采用一个数组 `matched_array[i]` 来记录已建立映射关系, 由图 2 可知, $|X_d^{i-\sigma} X_d^i|$ 表示了目的域中搜索的长度, $|X_r^{i-\sigma} X_r^i|$ 表示参考域中搜索的长度。由于主要的计算复杂度位于参考域的搜索过程, 因此搜索转移算法中 $|X_r^{i-\sigma} X_r^i|$ 的长度是提高计算效率的关键。若是仍依旧传统算法逐行选取目的像素, 会导致目的域逆序方向上无法找到最近相邻像素, 直至到达图像的边缘像素, 从而无法保证有效地获得 $X_d^{i-\sigma}$ 和降低 $|X_r^{i-\sigma} X_r^i|$ 的长度。针对该问题, 本文从空间连续性出发, 通过自适应调整目的域的绘制顺序来保证有效的计算。

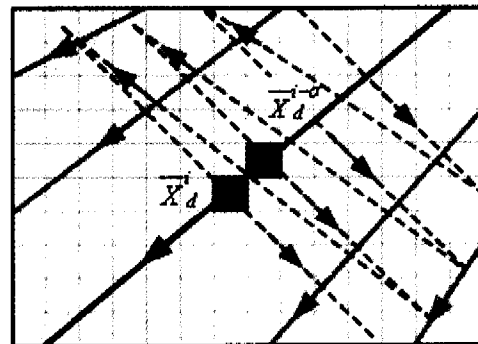


图4 相对极线的绘制顺序

由于参考域中的极线对于目的视点存在一个从远到近的排序^[4], 而在符合极线整体匹配性和保序性的场景中, 极线上的对应像素具有相同的排列, 因此在目的域中同样存在从远到近的排序。为了控制 $|X_r^{i-\sigma} X_r^i|$ 的长度, 需要尽量减少 $X_r^{i-\sigma}$ 和 X_r^i 对应点的空间距离。由于 $f(X_r^{i-\sigma}) = X_d^{i-\sigma}$ 且 $f(X_r^i) =$

\bar{X}_i^o ,因此关键在于如何确定 \bar{X}_i^o ,使得空间对应点最近。如图4,在理想情况下,沿极线方向,当 \bar{X}_i^o 和 \bar{X}_i^r 空间相邻($\sigma=1$)时, $|\bar{X}_i^o - \bar{X}_i^r|$ 具有最小值。同时,为了保证在对 \bar{X}_i^o 进行匹配时,其对应的相邻最近像素 \bar{X}_i^o 已经得到匹配,目的像素的遍历设定为按照极线方向横向扫过极线的顺序进行(如图4中虚线所示)。这样,位于同一极线上的像素会沿着极线方向顺序匹配。

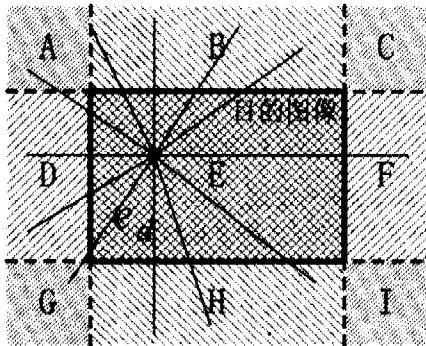


图5 极点位于成像空间

因此,根据目的极点 $e_d = [u_d^o \ v_d^o \ w_d^o]$ 落在目的成像平面上的相对位置,本文算法中,我们将成像面划分为9个区域(图5),对于位于不同区域的极点采用不同的绘制顺序,以保证能最快地在目的域中匹配到最近相邻像素。根据极点的位置和极线传播的方向,目的域绘制顺序如图6所示。

- (1)若 e_d 位于 A,C,G,I 区,绘制采用 ZigZag 顺序遍历所有目的像素点;
- (2)若 e_d 位于 B,D,F,H 区,根据各自位置逐行或逐列进行;
- (3)若 e_d 落在目的图像 E 内,则根据具体位置采用蛛网顺序进行遍历。

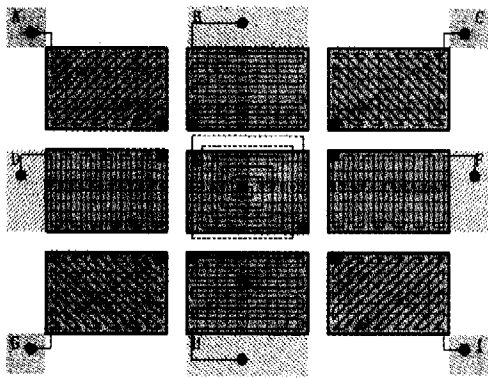


图6 目的像素的遍历顺序

3.4 最近相邻像素

最近相邻像素的作用在于为匹配建立新的起点,从而简化参考域匹配过程。但根据确定目的域的绘制顺序,沿着目的极线逆序搜索到的首个匹配像素并非都能简化参考域的计算,因此针对映射 $f(\bar{X}_i^o) = \bar{X}_i^r$ 具体分以下两种情况讨论:

- (1)目的极点位于目的图像区域外。此时极线在参考域内单向分布,只需沿着目的极线的逆序进行搜索。若存在首个已匹配像素 \bar{X}_i^r ,则记为最近相邻像素 $\bar{X}_i^o = \bar{X}_i^r$;若搜索到图像边缘像素,则标记当前目的像素无最近相邻像素 $\bar{X}_i^o = \phi$ 。
- (2)目的极点位于目的图像区域内。此时极线分布会在

目的极点处改变方向,记目的极点 $e_d = [u_d^o \ v_d^o \ w_d^o] = P_d^{-1}(C_r - C_d)$ 。由极线几何原理可知:当 $w_d^o > 0$ 时,极线在极点两侧呈发散方向,因此沿极线逆序搜索时,若 $\bar{X}_i^o \in \overline{X_i^o e_d}$,即匹配到的像素位于目的像素和极点之间,则 \bar{X}_i^o 对应空间近点,可以作为最近相邻像素 $\bar{X}_i^o = \bar{X}_i^r$;若 $\bar{X}_i^o \in \overline{X_i^o e_d}$,即搜索中经过了极点,使得 \bar{X}_i^o 和 \bar{X}_i^r 处的极线分布方向不一致,由于此时 e_d 对应空间最近点,因此若 e_d 已匹配,则将极点记为最近相邻像素 $\bar{X}_i^o = e_d$,否则 $\bar{X}_i^o = \phi$ 。而当 $w_d^o < 0$ 时,极线在极点两侧呈收敛方向,逆序搜索时不会产生变相。

3.5 参考域搜索

在确定了 \bar{X}_i^o 的最近相邻像素 \bar{X}_i^o 后,以 \bar{X}_i^o 为起点,搜索转移到参考域进行。在参考极线上进行搜索匹配时,为了保证目的域中正确的可见性,我们根据 list-priority 算法^[5]原理,使得参考像素相对于目的视点由近及远地进行验证,从而保证最先匹配的像素为最近可见点。记参考域中 $\bar{X}_i^r = [u_i^r \ v_i^r \ w_i^r] = P_r^{-1}(C_d - C_r)$, $\bar{X}_i^o = [u_i^o \ v_i^o \ w_i^o] = P_r^{-1}P_d$,本文算法参考域搜索方向为:若 $w_i^o > 0$ 且 $w_i^r > 0$, \bar{X}_i^o 对应空间最近点,搜索从 \bar{X}_i^o 向 \bar{X}_i^r 进行;若 $w_i^o > 0$ 且 $w_i^r < 0$, \bar{X}_i^o 反相对应空间最远点,搜索从 \bar{X}_i^o 向 \bar{X}_i^r 进行;若 $w_i^o < 0$ 且 $w_i^r > 0$, \bar{X}_i^o 对应空间最远点,搜索从 \bar{X}_i^o 向 \bar{X}_i^r 进行;若 $w_i^o < 0$ 且 $w_i^r < 0$,参考域中无匹配点,将 \bar{X}_i^o 标记为空洞。

需要说明的是,由于部分目的像素不存在最近相邻像素,搜索需要恢复到传统算法的极线复杂度。但采用定制的绘制顺序后,这部分像素主要为图像的边缘像素,其匹配完成后往往作为其他目的像素的最近相邻点,由于数量有限,因此不会显著影响算法的效率。

3.6 算法总结

综上所述,搜索转移算法表示为:

```
SearchTrans( $R_r, R_d$ )
{
  ComputeDesiredEpipole( $C_r, C_d, P_d$ ); //计算目的域极点  $e_d$ 
  SetWarpingOrder( $e_d$ ); //根据  $e_d$  设置目的域遍历顺序
  for every desired pixel in order ( $\bar{X}_i^o$ ) //按照顺序依次取目的像素  $\bar{X}_i^o$ 
  {
    ComputeDesiredEpipolarLine( $\bar{X}_i^o$ ); //计算  $\bar{X}_i^o$  对应的目的极线  $l_i^o$ 
    for pixel  $\bar{X}_i^r \in l_i^o$  in inverse order
    {
      if (marched_array[ $\bar{X}_i^r$ ] = NULL) continue; }  $\bar{X}_i^o = \bar{X}_i^r$ ; //目的域搜索最近相邻像素
    ComputeReferenceEpipolarLine(); //计算  $\bar{X}_i^o$  对应的参考极线  $l_i^r$ 
    ComputeReferenceSegment( $l_i^r, \bar{X}_i^o$ ); //计算有效搜索线段及方向 for pixel  $\bar{X}_i^r$  in segment
    {
      ComputeNeededDepth(); //计算需要的普遍视差  $\delta_{needed}$ 
      if ( $\delta_r(\bar{X}_i^r) > \delta_{needed}$ )  $\bar{X}_i^o = \bar{X}_i^r$ ; //当对应点普遍视差满足,搜索结束
      RecordMarchPoint(marched_array); //将当前匹配关系记录在 marched_array 中
    }
  }
  FillHole(); //从其他参考域填补空洞
}
```

4 算法性能检验

4.1 复杂度分析

我们从计算复杂度的角度对本文算法和传统算法的性能进行检验。传统算法中,建立单次映射匹配的复杂度由参考像素验证匹配计算和进行验证的次数决定,即 $Cost_1 = [O(ComputeNeededDepth()) + O(if())] \times d_d^1$,其中 d_d^1 表示在参考域搜索像素的个数;而本文算法的复杂度 $Cost_2 = O(ComputeNeededDepth()) \times d_r^2 + O(if()) \times d_d^2$,其中 d_d^2 表示目的域搜索像素个数, $d_r^2 = |\bar{X}_i^o - \bar{X}_i^r| = \sigma$, $ComputeNeededDepth()$ 是计算需要的普遍视差,具体计算公式可由式(1)推导展开,表示为:

$\delta_{need} =$

$$\frac{(\vec{a}_r \cdot (\vec{b}_d \times \vec{c}_d))u_r^i + (\vec{b}_r \cdot (\vec{b}_d \times \vec{c}_d))v_r^i + (\vec{c}_r \cdot (\vec{b}_d \times \vec{c}_d)) - u_d^i [(\vec{a}_r \cdot (\vec{a}_d \times \vec{b}_d))u_r^i + (\vec{b}_r \cdot (\vec{a}_d \times \vec{b}_d))v_r^i + (\vec{c}_r \cdot (\vec{a}_d \times \vec{b}_d)]}{((\vec{C}_r - \vec{C}_d) \cdot (\vec{a}_d \times \vec{b}_d))u_d^i - ((\vec{C}_r - \vec{C}_d) \cdot (\vec{b}_d \times \vec{c}_d))} \quad (3)$$

式中参考和目的像素分别为 $\vec{X}_r^i(u_r^i, v_r^i)$ 和 $\vec{X}_d^i(u_d^i, v_d^i)$, 参考和目的相机矩阵分别表示为 $P_r = [\vec{a}_r \ \vec{b}_r \ \vec{c}_r]$ 及 $P_d = [\vec{a}_d \ \vec{b}_d \ \vec{c}_d]$. 这样, 一次 $O(\text{ComputeNeededDepth}())$ 计算需要进行 9 次乘法和 6 次加法操作, 与之相比 $O(\text{if}())$ 可以忽略不计. 因此

$$\text{Cost}_1 / \text{Cost}_2 \approx d_r^i / d_d^i = d_r^i / \sigma$$

即算法复杂度的比例由参考域搜索的长度决定. 由于 d_r^i 是图像分辨率相关的 $d_r^i = O(n)$ (设图像分辨率为 $n \times n$). 而 σ 表示了极线上空间相邻点间的像素个数, 远远小于 n (实验统计表明 $\sigma < 2 < n$), 因此大大降低了计算复杂度.

4.2 实验比较

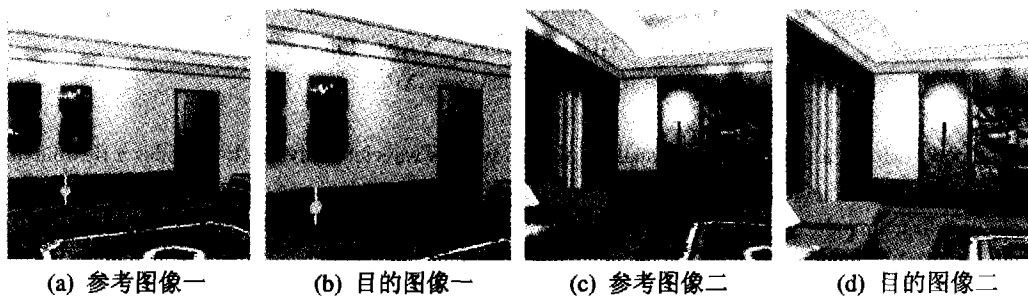


图 7 部分实验参考和结果图像

表 1 实验数据统计表

	传统算法			本文算法			
	\bar{d}_1	参考域搜索时耗	算法时耗	\bar{d}_2	目的域搜索时耗	参考域搜索时耗	算法时耗
test1	204.651	24.110	24.854	1.373	<0.001	1.342	2.234
test2	221.218	25.147	25.895	1.520	<0.001	1.545	2.435
test3	186.725	22.069	22.812	1.091	<0.001	1.216	2.103
test4	198.013	22.896	23.643	1.225	<0.001	1.322	2.210
平均	202.652	23.555	24.301	1.302	<0.001	1.356	2.246
搜索复杂度降低率	99.36%			提速率	1081.97%		

由表 1 可知, 传统图像逆映射算法中, 由于参考域匹配搜索的复杂性, 参考域搜索的时耗占算法整体时耗的 95% 以上, 从而极大地约束了算法效率的提高. 而采用搜索转移算法后, 大量的搜索匹配过程被转移到目的域, 使得参考域的搜索时耗从平均 23.555s 降到了 1.356s. 与此同时, 由于目的域中的搜索匹配只是简单的判断语句, 其平均时耗几乎可以忽略不计 (少于 0.001s), 因而本文算法可以大大提高算法整体的计算效率. 实验数据表明, 本文算法将参考域的平均实际搜索复杂度降低了 99.36% (平均从 202.652 降低到 1.356), 提高计算效率 10 倍左右 (平均从 24.301s 降到 2.246s).

结论 本文设计了一种图像逆映射的快速算法—搜索转移算法, 该算法原理简单, 易于实现, 加速效果明显. 利用场景特性, 我们还可以从更多的方面进行对图像映射算法进行优化, 如利用极线的整体匹配性在目的域进行逐极线的增量匹配, 从而更好地利用连续性以提高单次匹配的速度; 或是根据不同的深度划分区域从而逐块逐区域的进行匹配, 以降低需要建立匹配的次, 这也是我们下一步研究的目标.

参考文献

1 Mark W R, McMillan L, Bishop G. Post-Rendring 3D warping [C]. In: Proceedings of the 1997 Symposium on Interactive 3D

我们在在一台 P4 (1.7G/384M) 的 PC 上采用 VC6.0 (OpenGL/GLUT) 实现了本文算法. 实验采用的参考和目的图像的分辨率均为 512×512 , 无需进行预处理. 我们针对一个室内场景在不同的参考和目的视点下采用传统算法和本文算法进行多次随机测试, 部分映射结果如图 7 所示. 我们对不同算法映射过程中的时耗以及搜索复杂度进行了统计, 结果如表 1 所示. 其中我们采用平均搜索像素数 $\bar{d} = \sum_{i=0}^{n \times n} d^i / n^2$ (d^i 为单次搜索像素个数) 来表示实际搜索复杂度.

Graphics, New York: ACM Press, 1997. 7~16
 2 Shade J, et al. Layered Depth Images [C]. In: Proceedings of the SIGGRAPH'98, 1998. 231~242
 3 Rafferty M M, Aliaga D G, Lastra A A. 3D Image Warping in Architectural Walkingsroughs [C]. VRAIS'98, 1998. 228~233
 4 McMillan L. An image-based approach to three-dimensional computer graphics; [Ph D Thesis]. University of North Carolina, 1997
 5 McMillan L, Bishop G. Plenoptic Modeling: an Image-based Rendering System [A]. Proceedings of the SIGGRAPH'95, 1995. 39~46
 6 Chang L W. Combining Multiple Reference Images in an Inverse Warper; [Master D Thesis]. MIT, 1998
 7 Laveau S, Faugeras O D. 3-D Scene Representation as a Collection of Images [C]. In: IEEE ICPR'94, 1994(A); 689~691
 8 Marcato R W. Optimizing an Inverse Warper; [Master [D] Thesis]. MIT, 1998
 9 Schaufler G, Stuerzlinger W. Three Dimensional Image Cache for Virtual Reality. Computer Graphics Forum (Eurographics'96), 1996, 15(3); 227~230
 10 马颂德, 张正友. 计算机视觉—计算机理论与算法基础[Z]. 北京: 科学出版社, 1998
 11 唐荣锡, 汪嘉业, 彭群生. 计算机图形学教程[Z]. 北京: 科学出版社, 1990