

光场图像透视变换算法

王建明^{1,2}, 毛一鸣^{1,2}, 晏涛^{1,2*}, 刘渊^{1,2}

¹江南大学数字媒体学院, 江苏 无锡 214122;

²江苏省媒体设计与软件技术重点实验室, 江苏 无锡 214122

摘要 提出一种基于全变分投影和空洞修复的光场图像透视变换算法。该算法可估计输入光场图像每个子视点的视差图,通过图像反向投影和重投影优化生成指定虚拟相机位置和姿态的光场图像,接着使用基于视差优先级的图像修复方法修复透视变换后的中心子视点图像,再通过中心子视点向水平和垂直方向进行内容传播,依次修复其他子视点图像。实验结果表明,所提方法生成的透视变换光场图像满足需求,可正确填充被遮挡区域,可以应用于多种光场图像编辑。

关键词 图像处理; 光场图像; 透视变换; 图像修复; 全变分; 光场图像编辑

中图分类号 TP391 文献标识码 A

doi: 10.3788/LOP56.151003

Perspective Transformation Algorithm for Light Field Image

Wang Jianming^{1,2}, Mao Yiming^{1,2}, Yan Tao^{1,2*}, Liu Yuan^{1,2}

¹ School of Digital Media, Jiangnan University, Wuxi, Jiangsu 214122, China;

² Jiangsu Key Laboratory of Media Design and Software Technology, Wuxi, Jiangsu 214122, China

Abstract This study proposes a perspective transformation algorithm for light field image based on total variational projection and hole restoration. The proposed method estimates disparity maps for all subaperture views of the input light field image. The light field image with the specified posture in the specified virtual camera position is generated via back projection and reprojection optimization. Using the image restoration algorithm based on the disparity priority, we restore the central subaperture view image after perspective transformation. The central subaperture views are used for content distribution in horizontal and vertical directions, and thus the rest subaperture view images are restored orderly. Experimental results show that the perspective transformed light field images generated by the proposed method meet requirements, and the covered regions are filled correctly. The proposed method can be applied to a series of light field image editing.

Key words image processing; light field image; perspective transform; image restoration; total variation; light field image editing

OCIS codes 100.2000; 100.6890; 100.4994; 110.4190

1 引言

透视变换是将图像投影到一个新的视平面,广泛应用于图像透视矫正^[1]、全景图像拼接^[2-3]等图像处理领域。光场图像记录场景丰富的三维结构和颜色信息,相较于传统的二维图像,可更好地用于估计场景的深度和进行三维重建或计算高质量的透视变

换结果。与光场图像的新视点插值算法相比,透视变换的视点变化范围更大,但会出现被遮挡区域变为可见区域的内容填充问题。光场图像透视变换需要考虑新相机姿态下每个子视点之间的连续和一致性,因此更具有挑战性。

传统双目立体图像、多视点立体图像的透视变换算法已经较为成熟。Du 等^[4]提出基于双目立体图像

收稿日期: 2019-01-31; 修回日期: 2019-02-28; 录用日期: 2019-03-05

基金项目: 国家科技支撑计划课题(2015BAH54F01)、国家自然科学基金(61872166)、江苏省自然科学基金(BK20170197)

* E-mail: yantao.ustc@gmail.com

的透视变换算法,通过对左右图像特征点进行匹配和对对应的线段进行透视变换,可以防止局部特征扭曲,同时避免产生空洞区域;Zhang等^[5]提出了一种双目立体图像生成方法,其光场图像生成满足视点之间轴距要求,可以根据需求生成满足视差/深度要求的双目立体图像;Chaurasia等^[6-7]使用稀疏的深度信息进行多视点图像透视变换,通过人为指定物体的轮廓或者根据颜色信息,将图像分成均匀的超像素保持物体的局部轮廓。相比双目立体图像、多视点立体图像,光场图像透视变换需要考虑更多的约束条件,因此传统的透视变化算法已不再适用于光场图像透视变换。

光场相机感光元件总像素数有限,其子透镜数量越多,单张图像的空间分辨率越小,即需要在光场图像的角度分辨率和空间分辨率之间进行权衡,目前针对光场图像的算法主要集中在新视点插值和图像超分辨率方面。Wanner等^[8]提出的全变分框架实现了光场图像的超分辨率新视点生成,但新视点的质量严重依赖于视差图的精度和图像本身的质量;Pujades等^[9]提出了一个基于启发式方法和贝叶斯公式的视点生成模型,由于考虑了场景几何估计不确定性所引起的误差,因此新视点图像的质量显著提高;Kalantari等^[10]提出了一种针对消费级光场相机新视点生成的神经网络算法,该算法同时训练视差和颜色两部分,只需要光场图像顶点4个子视点图像,即可生成中间其他子视点图像;Flynn等^[11]提出了基于深度卷积神经网络(DCNN)的多视点图像新视点生成算法,将新视点生成问题分解为多个深度平面上的视点选择和像素颜色融合问题,在谷歌街景数据上进行训练和预测,得到了良好的结果;Gul等^[12]提出的基于卷积神经网络的光场图像增强算法包含两个子网络,分别用来提高角度分辨率和空间分辨率,适用于消费级光场相机;Wang等^[13]利用端到端的深度学习框架,构建了一个伪四维(4D)的卷积神经网络,通过稀疏的视点输入即可高效合成密集的4D光场。上述算法可以实现光场图像内部的视点插值效果和子视点图像的超分辨率,但无法生成光场图像外部指定相机姿态的透视变换光场图像。

针对现有算法的不足,本文提出一种基于全变分投影和空洞修复的光场图像透视变换算法。该算法包括两部分:1)通过子图像的透视变换优化生成目标光场图像;2)利用视差值判断目标光场图像空洞区域的内容填充顺序。透视变换过程需要考虑场景的遮挡关系,一个子视点中多个点投影到目标子视点图像上同一个点时,通过深度信息即可判断点

与点之间的遮挡关系。光场图像空洞区域的内容填充问题难点在于修复顺序,需要考虑场景的层次关系以及如何避免前景纹理填充背景区域。通过图像的视差值判断修复区域边缘属于前景还是背景,进而判断修复的优先级,确保内容填充顺序正确。最后,通过光场图像子视点之间的内容传播保持子视点之间的连续和一致性。

2 光场图像透视变换算法原理

首先,对光场图像进行预处理,得到每个子视点的视差图;然后,对光场图像进行基于全变分优化的透视变换,得到目标虚拟相机位置和姿态的光场图像以及对应的视差图;接着,使用基于视差优先级的方法修复视差图和中心子视点图像;最后,通过中心子视点向水平和垂直方向依次修复其他子视点图像。

2.1 基于全变分的光场图像透视变换

假设输入的光场图像为 $\sum_{s=1, t=1}^{n_s, n_t} v_{s,t} : \mathcal{R}_{s,t} \rightarrow \mathbb{R}$, $s \in [1, n_s], t \in [1, n_t]$, 其中 n_s 为水平方向子视点的数量, n_t 为垂直方向子视点的数量, s, t 分别为水平和垂直方向子视点的索引, $\mathcal{R}_{s,t}$ 为输入相机所在平面, $:$ 表示视点图像 $v_{s,t}$ 是相机平面到像素范围实数集的映射。该算法的目标是给定一个旋转角度和平移量,生成虚拟相机姿态下的光场图像 $\sum_{s=1, t=1}^{n_s, n_t} u_{s,t} : \Pi_{s,t} \rightarrow \mathbb{R}$, $s \in [1, n_s], t \in [1, n_t]$, $\Pi_{s,t}$ 为目标相机所在平面,目标光场图像生成过程可用一个全变分优化模型表示,即

$$E(u_{s_u, t_u}) = \int_{\Pi_{s_u, t_u}} |Du| + \sum_{s_v=1, t_v=1}^{n_s, n_t} \frac{1}{2} \int \mathcal{R}_{s_v, t_v} \times m_{s_v, t_v} (u_{s_u, t_u} \circ \tau_{s_v, t_v} - v_{s_v, t_v})^2 dx \quad (1)$$

式中: E 为能量函数; s_u, t_u 分别为目标视点水平、垂直方向的索引; D 为二维平滑函数; u 为视点 u_{s_u, t_u} 的图像矩阵; Π_{s_u, t_u} 表示目标相机所在平面; s_v, t_v 分别为输入视点水平、垂直方向的索引; \circ 表示透视变换符号; $m_{s_v, t_v} \in \{0, 1\}$ 表示目标光场图像子视点 u_{s_u, t_u} 和输入光场图像子视点 v_{s_v, t_v} 的遮挡关系,当且仅当视点 v_{s_v, t_v} 中一个点在视点 u_{s_u, t_u} 中可见时, $m_{s_v, t_v} = 1$, 否则 $m_{s_v, t_v} = 0$; τ_{s_v, t_v} 表示视点 u_{s_u, t_u} 到视点 v_{s_v, t_v} 的透视变换,为了计算方便,引入 τ_{s_v, t_v} 的逆变换 $\gamma_{s_v, t_v} = (\tau_{s_v, t_v} | v_{s_v, t_v})^{-1}$, 表示从视点 v_{s_v, t_v} 到视点 u_{s_u, t_u} 的透视变换, γ_{s_v, t_v} 可以通过反向投影和重

投影两个步骤计算得到。

反向投影是从二维图像到三维空间的映射,因此需要已知场景的深度信息,本文采用基于光场图像极平面图像(EPI)和结构张量的方法^[14]计算视差图。图像中的每个点 $p(x, y)$ 在空间中的坐标 $P(X, Y, Z)$ 可由双目相机模型^[15]计算得到,即

$$\begin{cases} X = \frac{Z}{f}(x - x_0) \\ Y = \frac{Z}{f}(y - y_0) \\ Z = \frac{bf}{d(x, y) + d_s} \end{cases}, \quad (2)$$

式中: b 为相邻两个子透镜之间的距离; f 为相机的等效焦距; x_0 和 y_0 分别为图像中心像素坐标和原点的偏移; $d(x, y)$ 为点 $p(x, y)$ 的视差; d_s 为相邻两个子视点之间的偏移。

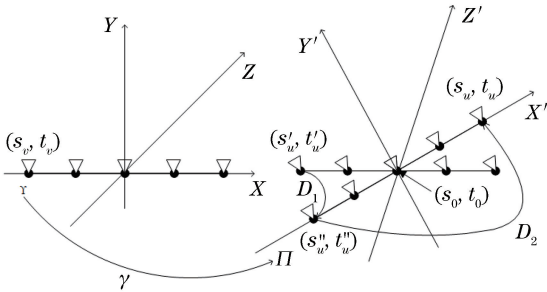


图1 光场图像透视变换模型

Fig. 1 Perspective transformation model for light field image

经过反向投影得到场景空间坐标信息后,即可计算场景空间坐标到目标光场图像的重投影。光场相机的所有子透镜位于同一个平面,相机姿态的变化等效于子透镜平面的变化。相机旋转时,中心点外的子透镜位置相对相机中心会产生一个平移,如图1中 D_1 所示,且目标子视点与参考子视点索引不一致时,两个子视点的位置也存在一个偏移,



图2 带空洞的光场图像透视变换结果。(a)输入的中心子视点图像;(b)单子视点图像透视变换结果;(c)全部子视点图像透视变换融合结果

Fig. 2 Perspective transformation results of light field image with black holes. (a) Input central subaperture view image; (b) perspective transformation result of single subaperture view image; (c) perspective transformation fusion result of all subaperture view images

如图1中 D_2 所示。因此,重投影过程需要考虑两部分的平移量,则有

$$\begin{cases} P' = R \cdot P + T_{\text{trans}} + (I - R) \cdot D_1 + D_2 \\ p' = F \cdot P' \end{cases}, \quad (3)$$

式中: F 为相机的内参矩阵, $F = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$; R 和

T_{trans} 分别为相机姿态变化的旋转矩阵和平移向量; P' 为目标相机坐标系下空间中的点; p' 为目标图像坐标系下的像素点; I 为全1矩阵; $D_1 = [s_c \ t_c \ 0]^T$ 为参考视点和中心视点位置的偏移; $D_2 = [s_t \ t_t \ 0]^T$ 为参考视点和目标视点位置的偏移,其参数计算方式为

$$\begin{cases} s_c = (s'_u - s_0) \times b \\ t_c = -(t'_u - t_0) \times b \\ s_t = (s''_u - s_u) \times b \\ t_t = -(t''_u - t_u) \times b \end{cases}, \quad (4)$$

式中: s_0 和 t_0 为光场图像中心子视点的索引; s_c, t_c 分别为参考视点相对于中心视点在水平、垂直方向的偏移; s_t, t_t 分别为参考视点相对于目标视点在水平、垂直方向的偏移; s'_u, t'_u 为不考虑相机平面的旋转时,透视变换图像所在视点水平、垂直方向的索引; s''_u, t''_u 为只考虑相机平面旋转但不考虑参考视点和目标视点位置的偏移时,透视变换图像所在视点水平、垂直方向的索引。

反向投影和重投影可将输入子视点 v_{s_v, t_v} 透视变换至目标子视点 u_{s_u, t_u} ,通过快速迭代阈值收缩算法(FISTA)^[16]最小化能量公式[(1)式]依次得到目标光场图像的所有子视点图像。透视变换中当输入子视点有多个点投影到目标子视点图像上同一个点时,通过这些点的深度信息判断多个点之间的遮挡关系。透视变换结果如图2所示,图2(a)所示为

输入的中心子视点图像;图 2(b)所示为单个子视点图像透视变换到目标子视点的图像;图 2(c)所示为所有输入视点透视变换到目标子视点后融合后的图像。结果表明,通过信息互补可消除不同输入子视点之间部分空洞区域。

2.2 基于视差优先级的图像修复方法

不同于图像编辑、物体去除等问题,透视变换的结果图像中大部分待修复区域由视点变化导致,即原本被前景遮挡的背景区域变成了可见区域。因此,只利用图像本身的纹理和颜色信息修复背景区域时^[17],容易错误引用前景的纹理信息。为了避免这种错误,采用先修复视差图后修复 RGB 图像的策略,通过视差图判断 RGB 图像空洞区域修复的优先级,得到相对准确合理的修复结果。

对于待修复的视差图 Γ ,假设待修复区域为 Ω ,已存在视差的区域为 Φ ,先提取待修复区域的边缘 ∂_Ω 。对于中心点为 $p \in \partial_\Omega$ 的图像块 Ψ_p ,修复优先级为

$$P(p) = \frac{\sum_{q \in \Psi_p \cap (\Gamma - \Omega)} C(q)}{|\Psi_p|}, \quad (5)$$

式中: q 为图像块中已存在的像素; $C(p)$ 表示点 p 的优先级。则有

$$C(p) = \begin{cases} 0, & p \in \Omega \\ 1 - \frac{[d(p) - d_{\min}]^2}{(d_{\max} - d_{\min})^2}, & p \in \Gamma - \Omega \end{cases}, \quad (6)$$

式中: $d(p)$ 为点 p 的视差; d_{\min} 、 d_{\max} 分别为视差的最小值和最大值。一个点的视差越接近 d_{\max} ,其深度越小,因此属于图像的前景区域,拥有更小的修复优先级,反之亦然。修复过程中每次迭代需要找到优先级最大的图像块 Ψ_p ,然后在 Φ 内搜索最相似的 Ψ_q ,使得

$$\Psi_p = \arg \min_{\Psi_q \in \Phi} \text{diff}(\Psi_p, \Psi_q), \quad (7)$$

式中: $\text{diff}(\Psi_p, \Psi_q)$ 表示两个图像块之间的差异,定义为两个图像块已存在像素的平方和,找到最相似的图像块后需要更新视差图以及优先级。则有

$$\begin{cases} \Psi_p = \Psi_q, & p \in \Psi_p \cap \Omega \\ C(p) = \alpha C(q), & p \in \Psi_p \cap \Omega, q \in \Psi_q \end{cases}, \quad (8)$$

式中: α 为递减系数,防止出现沿某个高优先级的方向一直修复的情况,本文实验中, $\alpha = 0.9$ 。当前图像块修复完成后,更新待修复区域的边缘,并寻找下一个最高优先级的待修复图像块,直至整个视差图像修复完成。

为了使修复的视差图像更准确,采用低秩低梯度的方法^[18]优化修复的视差图像,将视差值量化为灰度级,然后进行低阶正则化,减少梯度为 1 的惩罚,去除视差图噪声的同时允许连续的视差变化。

RGB 图像修复和视差图像修复步骤相似,过程如下:

- 1) 计算图像每个点的优先级 $C(p) = 1 - \frac{[d(p) - d_{\min}]^2}{(d_{\max} - d_{\min})^2}$;
- 2) 提取图像待修复区域的边缘 ∂_Ω ;
- 3) 计算中心点 p 位于边缘 ∂_Ω 上图像块 Ψ_p 的修复优先级 $P(p)$;
- 4) 找到优先级最大的待修复图像块 Ψ_p ,在图像范围内寻找最相似的图像块 Ψ_q ;
- 5) 将 Ψ_q 的颜色信息复制到 Ψ_p 中对应的点;
- 6) 转到步骤 2),直至图像中所有的待修复区域修复完成。

光场图像修复结果如图 3 所示,修复后的图像前后景关系正确,纹理信息自然。



图 3 基于视差优先级的光场图像修复结果。(a)待修复的子视点图像;(b)待修复的子视点视差图;
(c)修复后的子视点图像;(d)修复后的子视点视差图

Fig. 3 Light field image restoration results based on disparity priority. (a) Subaperture view image to be restored; (b) subaperture disparity map before restoration; (c) restored subaperture view image; (d) restored subaperture disparity map

2.3 通过中心子视点修复其他子视点图像

采用基于空域的方法修复整个光场图像,每个子视点图像寻找最相似图像块的过程存在随机性,因此,非常容易出现子视点之间不一致的现象,而光场图像需要保持每个子视点之间均连续一致。因此,本文先修复中心子视点图像,然后依次修复其他子视点图像,

视差的连续性使得修复后的子视点之间内容保持一致。修复顺序为中心子视点图像向垂直方向修复中间一列子视点图像,如图4中实线箭头方向所示;再由每行中心子视点图像向水平方向依次修复其他子视点图像,如图4中虚线箭头方向所示。结果表明,修复后的光场图像子视点之间连续一致。

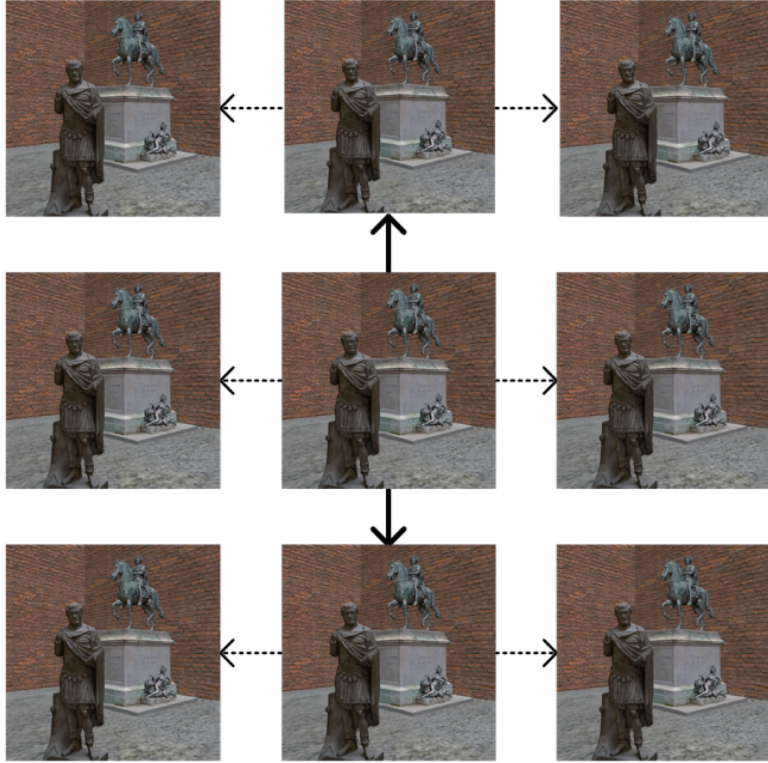


图4 光场图像所有子视点的修复顺序

Fig. 4 Restoration order of all subaperture views in light field image

修复某一个子视点图像时,将其最邻近一幅已修复的子视点图像作为参考图像。通过视差即可找到参考图像中点 p_{ref} 在待修复子视点图像中对应的点 $p = p_{\text{ref}} + d_{\text{ref}}$, d_{ref} 为参考图像的视差图,进而完成待修复子视点图像缺失的颜色信息填补。

3 实验结果

实验所用计算机硬件配置为 Intel(R) Core(TM) i7-6700k 处理器,主频为 4.00 GHz,内存为 16 GB,显卡为双 NVIDIA(R) GeForce(R) GTX 1070,操作系统为 Windows10,使用 Matlab 软件开发环境。仿真数据采用 Blender 软件进行渲染,真实数据通过 Lytro Illum 光场相机进行采集,使用 LFTToolBox^[19] 工具进行光场图像解码预处理。所有数据角度分辨率均为 9×9 ,即每组数据在水平和

垂直方向各有 9 个子视点,总共 81 幅子视点图像。子视点的索引通过二维向量表示,左上角和右下角的子视点索引分别为 $(0,0)$ 和 $(8,8)$ 。

3.1 透视变换结果分析

实验结果如图 5 所示,图 5 (a1)~(f1)为输入光场图像中心子视点图像;图 5 (a2)~(f2)为透视变换后的中心子视点图像,绿色部分为视点变换导致的空洞区域;图 5 (a3)~(f3)为修复后的中心子视点图像。第 1~3 行为 Blender 软件渲染的仿真数据;第 4~6 行为 Lytro 相机采集的真实数据,两者相比,由于后者子透镜之间的距离较大,因此空洞区域更明显。仿真实验的透视变换参数如表 1 所示,其中 b 为子透镜之间的距离, T_x 、 T_y 、 T_z 分别为沿着坐标轴 x 、 y 、 z 平移的距离, R_x 、 R_y 、 R_z 分别为绕坐标轴 x 、 y 、 z 旋转的角度。

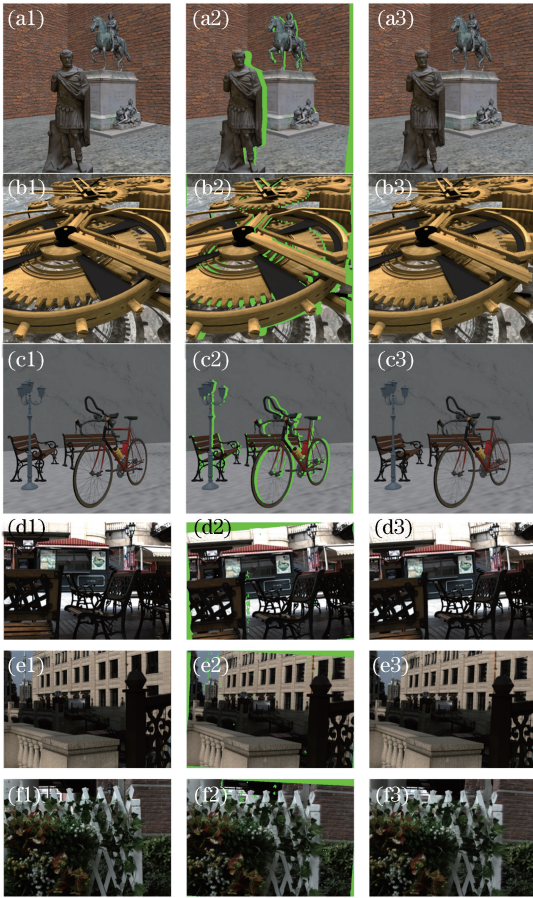


图5 光场图像透视变换及修复结果。(a1)~(f1)输入中心子视点图像;(a2)~(f2)透视变换结果;(a3)~(f3)修复后的透视变换结果

Fig. 5 Results of light field perspective transformation and restoration. (a1)-(f1) Input central subaperture view images; (a2)-(f2) perspective transformation results; (a3)-(f3) restored perspective transformation results

从图5可以看出,本文算法生成的光场图像

表1 透视变换参数

Table 1 Parameters for perspective transformation

Parameter	Fig. 5(a)	Fig. 5(b)	Fig. 5(c)	Fig. 5(d)	Fig. 5(e)	Fig. 5(f)
b / mm	1.00	1.00	1.50	1.12	1.65	0.86
T_x / mm	-20	-20	-30	-60	-45	-50
T_y / mm	0	0	0	0	0	0
T_z / mm	0	0	0	0	0	0
$R_x / (^\circ)$	0	0	0	1	0.5	0.5
$R_y / (^\circ)$	0	0.5	0.5	1	0.5	0.5
$R_z / (^\circ)$	0	0	0	5	-3	-3

表2 算法消耗时间

Table 2 Consuming time of proposed algorithm

Parameter	Fig. 5(a)	Fig. 5(b)	Fig. 5(c)	Fig. 5(d)	Fig. 5(e)	Fig. 5(f)
T / s	283.98	286.95	305.79	302.98	298.11	282.24

具有明显的透视变换效果,如图5(a)中雕塑、图5(b)中钟表齿轮以及图5(c)中路灯和自行车所示。修复后的纹理也正确考虑了前后景的遮挡关系,如图5(a)中石墩纹理延伸和图5(b)中钟表突起移动部分的被遮挡区域填充等所示。算法消耗时间如表2所示,表中数据为渲染单个子视点图像所需的时间。

3.2 透视变换结果对比

基于贝叶斯的新视点生成算法得到的结果和本文算法结果如图6所示,其中,图6(a1)~(c1)为文献[8]算法结果;图6(a2)~(c2)为文献[9]算法结果;图6(a3)~(c3)为本文算法结果;图6(a4)~(c4)为图像真值。

从图6可以看出,透视变换的范围较大时,新视点生成算法所得结果中物体边缘存在重影现象,如图6(a)中雕像、图6(c)中自行车和路灯所示,同时结果图像存在模糊现象和较大的噪声,如图6(a)中雕像头部、图6(b)中钟表边缘等所示。本文算法结果与真值相近,没有出现明显的纹理错误或图像噪声。计算透视变换图像的峰值信噪比(PSNR)和结构相似性(SSIM),定量分析和比较本文算法和文献[8-9]算法,结果如表3所示。可以看出本文算法的PSNR和SSIM值都明显高于文献[8-9]算法结果,透视变换结果具有更高的准确度和更好的视觉效果,充分证明了本文算法的有效性。

4 结论

提出了一种基于全变分投影和空洞修复的光场图像透视变换算法,通过子视点图像的反向投影和重投影优化生成目标光场图像,利用图像的视差判

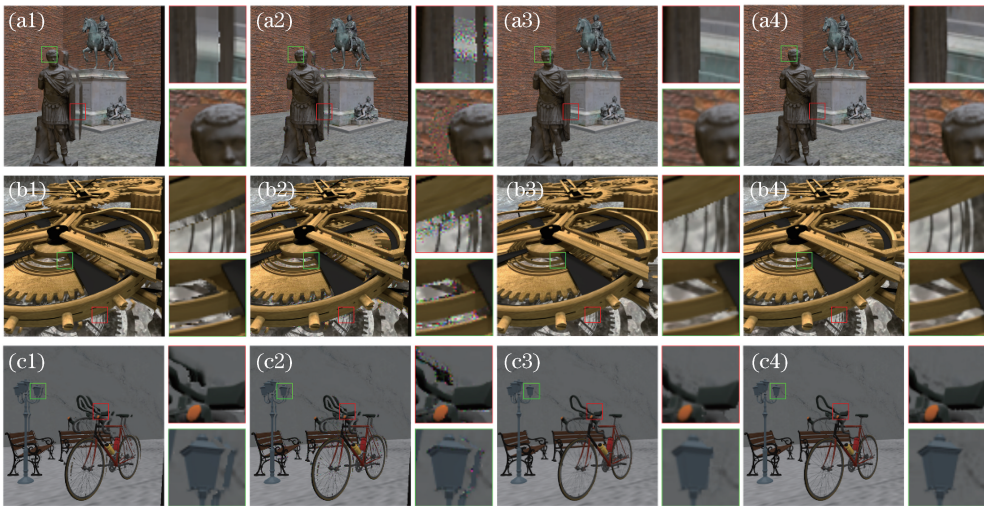


图6 光场图像透视变换结果比较。(a1)~(c1)文献[8]算法结果;(a2)~(c2)文献[9]算法结果;
(a3)~(c3)本文算法的结果;(a4)~(c4)图像真值

Fig. 6 Comparison of light field image perspective transformation results. (a1)-(c1) Results of method in Ref. [8];
(a2)-(c2) results of method in Ref. [9]; (a3)-(c3) results of proposed method; (a4)-(c4) truth images

表3 光场透视变换结果定量分析

Table 3 Quantitative analysis for light field perspective transformation images

Parameter	Method	Fig. 6(a)	Fig. 6(b)	Fig. 6(c)
PSNR	Method in Ref. [8]	23.18	22.38	22.72
	Method in Ref. [9]	26.35	22.06	22.53
	Proposed	35.43	25.69	30.74
SSIM	Method in Ref. [8]	0.67	0.81	0.79
	Method in Ref. [9]	0.78	0.81	0.79
	Proposed	0.96	0.89	0.95

断空洞区域的边缘属于前景还是背景,进而计算修复优先级。该方法克服了透视变换产生的空洞区域难填充问题,生成的透视变换光场图像拥有正确的遮挡关系,可服务于一系列光场图像编辑应用,如光场图像透视矫正、光场显示、光场图像拼接等。后续工作将针对本文算法的不足进行改进,例如,真实场景的透视变换结果受视差图的质量限制,可以采用更多的约束提高算法的稳健性,同时提高算法的优化速度,使算法具有实时性;另外,透视变换光场图像目前只能保持原来的分辨率,后续工作将在透视变换的同时使图像达到超分辨率,进一步提高算法实用性。

参 考 文 献

[1] Dai Q, Wang Y J, Han G L. Perspective image rectification based on improved Hough transformation and perspective transformation [J]. Chinese Journal of Liquid Crystals and Displays, 2012, 27(4): 552-556.

代勤,王延杰,韩广良.基于改进 Hough 变换和透视变换的透视图像矫正[J].液晶与显示,2012,27(4):552-556.

[2] Zhao T, Kang H L, Zhang Z P. Fast image mosaic algorithm based on area blocking and BRISK [J]. Laser & Optoelectronics Progress, 2018, 55(3): 031005.
赵婷,康海林,张正平.结合区域分块的快速 BRISK 图像拼接算法[J].激光与光电子学进展,2018,55(3):031005.

[3] Chen J H, Guo W S. Method of panoramic image stitching for theodolite-camera system [J]. Laser & Optoelectronics Progress, 2016, 53(5): 051001.
陈继华,郭文松.基于摄影经纬仪的全景图像拼接方法[J].激光与光电子学进展,2016,53(5):051001.

[4] Du S P, Hu S M, Martin R R. Changing perspective in stereoscopic images [J]. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(8): 1288-1297.

[5] Zhang L, Zhang Y H, Huang H. Efficient variational light field view synthesis for making

- stereoscopic 3D images [J]. *Computer Graphics Forum*, 2015, 34(7): 183-191.
- [6] Chaurasia G, Sorkine O, Drettakis G. Silhouette-aware warping for image-based rendering [J]. *Computer Graphics Forum*, 2011, 30(4): 1223-1232.
- [7] Chaurasia G, Duchene S, Sorkine-Hornung O, *et al.* Depth synthesis and local warps for plausible image-based navigation [J]. *ACM Transactions on Graphics*, 2013, 32(3): 30.
- [8] Wanner S, Goldluecke B. Spatial and angular variational super-resolution of 4D light fields[M] // Fitzgibbon A, Lazebnik S, Perona P, *et al.* *Computer vision-ECCV 2012. Lecture notes in computer science*. Berlin, Heidelberg: Springer, 2012, 7576: 608-621.
- [9] Pujades S, Devernay F, Goldluecke B. Bayesian view synthesis and image-based rendering principles[C] // 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 23-28, 2014, Columbus, OH, USA. New York: IEEE, 2014: 3906-3913.
- [10] Kalantari N K, Wang T C, Ramamoorthi R. Learning-based view synthesis for light field cameras [J]. *ACM Transactions on Graphics*, 2016, 35(6): 193.
- [11] Flynn J, Neulander I, Philbin J, *et al.* Deep stereo: learning to predict new views from the world's imagery[C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 5515-5524.
- [12] Gul M S K, Gunturk B K. Spatial and angular resolution enhancement of light fields using convolutional neural networks[J]. *IEEE Transactions on Image Processing*, 2018, 27(5): 2146-2159.
- [13] Wang Y L, Liu F, Wang Z L, *et al.* End-to-end view synthesis for light field imaging with pseudo 4DCNN[M] // Ferrari V, Hebert M, Sminchisescu C, *et al.* *Computer vision-ECCV 2018. Lecture notes in computer science*. Cham: Springer, 2018, 11206: 340-355.
- [14] Wanner S, Goldluecke B. Globally consistent depth labeling of 4D light fields[C] // 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 16-21, 2012, Providence, RI, USA. New York: IEEE, 2012: 41-48.
- [15] Yuan R F, Liu M, Hui M, *et al.* Depth map stitching based on binocular vision [J]. *Laser & Optoelectronics Progress*, 2018, 55(12): 121013. 袁瑞峰, 刘明, 惠梅, 等. 基于双目视觉的深度图拼接[J]. *激光与光电子学进展*, 2018, 55(12): 121013.
- [16] Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems [J]. *SIAM Journal on Imaging Sciences*, 2009, 2(1): 183-202.
- [17] Criminisi A, Pérez P, Toyama K. Region filling and object removal by exemplar-based image inpainting [J]. *IEEE Transactions on Image Processing*, 2004, 13(9): 1200-1212.
- [18] Xue H Y, Zhang S M, Cai D. Depth image inpainting: improving low rank matrix completion with low gradient regularization [J]. *IEEE Transactions on Image Processing*, 2017, 26(9): 4311-4320.
- [19] Dansereau D G, Pizarro O, Williams S B. Decoding, calibration and rectification for lenselet-based plenoptic cameras [C] // 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 23-28, 2013, Portland, OR, USA. New York: IEEE, 2013: 1027-1034.