

# 基于反向映射的元素图像生成算法

谷岳佳楠, 朴燕\*, 邓莉巾

长春理工大学电子信息工程学院, 吉林 长春 130022

**摘要** 传统方法是建立从光场到显示面的映射计算生成元素图像, 但会存在很多冗余映射。针对这一问题, 提出沿光路逆向迭代的算法, 建立从元素图像显示面到重建光场的单射, 使元素图像中的每个像素只对应唯一的光场像素, 这可以提高像素的匹配精度, 消除深度阶跃处的空洞。所提算法的核心是以元素图像显示面每一点为始点, 经过透镜的光心作射线, 通过迭代搜索求出射线与光场曲面距离观察者最近的交点, 并作为元素图像的匹配点。所提算法的时间复杂度主要受控于元素图像阵列的像素总和, 元素图像的计算生成速率是现有算法的 8 倍以上, 图像中的像素总数越大, 所提算法的速率优势越明显, 且实验结果验证理论推导的正确性。

**关键词** 成像系统; 集成成像; 微透镜阵列; 元素图像; 深度信息

中图分类号 TP391.4

文献标志码 A

doi: 10.3788/AOS202040.1911002

## Elemental Image Generation Algorithm Based on Reverse Mapping

Gu Yuejianan, Piao Yan\*, Deng Lijin

*College of Electronic and Information Engineering, Changchun University of Science and Technology,  
Changchun, Jilin 130022, China*

**Abstract** The traditional method is to establish a mapping calculation from the light field to the display surface to generate elemental images, but there will be a lot of redundancy. To solve this problem, an iterative algorithm along a mapped light path is proposed so that a single shot from an elemental image display surface to a reconstructed light field can be established. Each pixel in an elemental image only corresponds to a unique light-field pixel, which can improve the pixel matching accuracy and eliminate a depth void at a step. The core of the proposed algorithm is based on each point on a display surface of an elemental image as the starting point, passing through the optical center of a lens as a ray through an iterative search to find the intersection of the ray and light field surface closest to an observer. Such an intersection is used as the matching point of the elemental image. The time complexity of the proposed algorithm is mainly controlled by the sum of pixels of an elemental image array. Calculation and generation rates of elemental images using the proposed algorithm are usually more than eight times those of the existing algorithm. An advantage of the proposed algorithm is that the total number of pixels in the image is directly proportional to the speed of the algorithm. The experimental results verify the correctness of the theoretical statements emphasized in this study.

**Key words** imaging systems; integral imaging; micro-lens array; elemental image; depth information

**OCIS codes** 110.3010; 100.2000; 100.6890

## 1 引 言

在工业生产和商业应用等领域, 人们对 3D 显示的需求日益增长。集成立体成像是一种裸眼 3D 立体显示技术, 观察者无需借助任何辅助设备就可以观看 3D 成像, 因此其逐步成为视频及图像显示领域的研究热点之一<sup>[1-6]</sup>。集成成像系统主要分为

采集和重构两个主要部分。采集原始场景信息的环节, 使用微透镜阵列采集与 3D 场景对应的多视角 2D 图像, 即元素图像(EI); 重构光场(LF)的环节, 利用采集到的 EI 阵列和微透镜阵列再现 3D 光场, 即在空间光调制器或液晶显示器上显示 EI 阵列, EI 阵列发出的光线被微透镜阵列折射, 折射后的光线在空间的叠加以还原 3D 场景。故 EI 是采集

收稿日期: 2020-05-07; 修回日期: 2020-06-12; 录用日期: 2020-06-23

基金项目: 国家自然科学基金(60977011)、吉林省科技计划(20180201091GX)、吉林省机器视觉智能装备与智能检测科技创新中心资助(20180623039TC)

\* E-mail: piaoyan@cust.edu.cn

与重构环节的重要因素之一,直接影响再现 3D LF 的成像质量。

针对光学采集过程中存在的客观问题,如相机或透镜阵列的配置复杂以及信号串扰,目前集成成像的采集过程主要使用计算机辅助来完成,减少采集过程中所需相机或透镜的数量,使用计算机进行后期处理可以有效提高 EI 的成像质量。基于这一思想,部分研究者采用点追踪渲染(PRR)<sup>[7]</sup>和视点矢量渲染(VVR)<sup>[8]</sup>的方法建立 3D 模型,进而生成 EI 阵列。针对医学成像数据,Jang 等<sup>[9]</sup>提出了基于 GPU 和八叉树(Octree)的实时采集方法。Xing 等<sup>[10]</sup>基于反向射线追踪技术,提出了一种相对高效的计算机集成成像方法。但上述方法的处理时间较长,更适用于展现虚拟的立体场景,这在一定程度上限制了集成成像技术的市场化。针对这一缺陷,Li 等<sup>[11]</sup>提出了一种使用深度相机实时获取深度图和颜色图以生成 EI 的集成成像采集方法。2017 年,王琼华团队<sup>[12]</sup>提出了一种基于双目立体相机的实时集成成像拍摄系统,简化了传统系统的结构。Hong 等<sup>[13]</sup>提出了一种新的方法,建立混合的立体 3D 相机系统,采用混合反投影的方法生成集成图像,改善了部分遮挡的重现效果。近年来,依据较少的场景基础信息数据,采集场景信息是集成成像研究的热点和主要方向。

目前,基于深度信息计算生成 EI 是场景信息采集的主要解决方案之一<sup>[14-16]</sup>。现有的 EI 生成算法通常建立从 3D 光场到显示终端的函数映射,但生成算法存在冗余映射,导致该类算法的计算速率偏慢。EI 对应原始场景中深度阶跃的位置,存在空洞(无匹配点)的现象,因此 EI 质量偏低。针对此类问题,本文提出一种根据场景的 2D 图像和场景深度信息计算生成 EI 阵列(EIA)的快速算法。所提算法能够同时解决成像质量和生成速率两个问题,并给出详细的理论分析和实验结果。

## 2 深度调整与场景深度反转

使用 4 维向量数组 $[x \ y \ z \ c]$ 来表示 3D 场景像素。其中 $(x, y, z)$ 为场景像素坐标, $(x, y)$ 为场景对应 2D 图像的像素坐标,此处取整数, $c = c(x, y)$ 为 2D 图像中 $(x, y)$ 点的像素值。彩色数字图像中, $c$ 为一个 32 位整数。实验过程中建立 EI 显示面(EIP)与光场的对应关系,不需要对  $c$  进行 RGB(Red, Green, Blue)分解。深度值  $z$  可以通过双目立体匹配算法和深度相机等技术来获取。

由于受到光学设备性能的限制,集成成像技术对再现光场的深度范围有严格的要求。超出深度范围限制的光场产生图像混叠,所以计算生成 EI 阵列前,首先采用文献[17-18]的方法来计算重建光场的深度范围,并对场景的深度进行调整和反转,再根据新的深度来计算生成 EI 阵列。

设原始场景中任一点的深度为  $h$ ,再现光场中对应点的深度为  $z$ ,具体转换过程如下。

计算再现光场中中心深度平面(CDP)的深度  $d$ ,表达式为

$$d = fg/(g - f), \quad (1)$$

式中: $f$  为透镜焦距; $g$  为 EIP 与透镜阵列之间的间距。

再现光场中像素的大小  $P_1$ ,即中心深度平面上像素的大小,则  $P_1$  表达式为

$$P_1 = dP_D/g, \quad (2)$$

式中: $P_D$  为 EIP 上的像素直径。

再现光场的深度范围  $\Delta d$  为

$$\Delta d = 2dP_1/P_L = 2d^2P_D/gP_L, \quad (3)$$

式中: $P_L$  为透镜之间的间距,由此  $z$  的最大值与最小值可表示为

$$z_{\min} = d - \frac{\Delta d}{2} = \frac{fg}{g - f} \left[ 1 - \frac{fP_D}{(g - f)P_L} \right], \quad (4)$$

$$z_{\max} = d + \frac{\Delta d}{2} = \frac{fg}{g - f} \left[ 1 + \frac{fP_D}{(g - f)P_L} \right]. \quad (5)$$

则  $h$  与  $z$  的反转关系,可表示为

$$z = \frac{h - \min(h)}{\max(h) - \min(h)}(z_{\min} - z_{\max}) + z_{\max}. \quad (6)$$

实验讨论的光场可以理解为一个完成深度转换后,从最大深度平面到最小深度平面之间的有限曲面,且曲面的分辨率与场景的 2D 图像是一致的。

## 3 LF 到 EIP 的映射算法

光场建立的坐标系如图 1 所示,省略  $X$  轴。为了计算方便, $z$  的零点设定在透镜的光心位置处,透镜的光心为  $L(x_0, y_0, 0)$ 。设  $XY$  平面上 2D 图像的分辨率为  $m \times n$ ,则再现光场的深度为一个深度矩阵 $[z(x, y)]_{m \times n}$ 。

光场中的一点  $A(x, y, z)$  对应 EIP 上点  $I_A(x_A, y_A)$ , $(x_A, y_A)$  是  $I_A$  在 EIP 上的像素坐标, $z = z(x, y)$  为点 $(x, y)$ 的深度。

$A$  与  $I_A$  的关系可表示为

$$x_A = \left[ x_0 - \frac{g}{z}(xP_1 - x_0) \right] \frac{1}{P_D}, \quad (7)$$

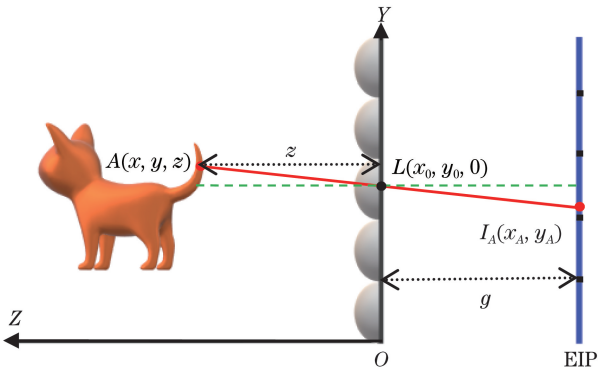


图 1 LF-EI 算法的 EI 生成原理

Fig. 1 EI generation principle of LF-EI algorithm

$$y_A = \left[ y_0 - \frac{g}{z}(y_{P1} - y_0) \right] \frac{1}{P_D}. \quad (8)$$

LF-EI 算法的原理:针对每个虚拟透镜  $L$ ,通过(7)式和(8)式可以求解光场中任意一点  $A(x, y, z)$  对应 EIP 上的位置  $I_A(x_A, y_A)$ ,从而得到透镜  $L$  对应的 EI。

经过水平相邻的两个透镜  $L1$  和  $L2$  后,点  $A$  对应 EIP 上的  $I_1(x_{L1}, y_{L1})$  和  $I_2(x_{L2}, y_{L2})$  的  $x$  坐标关系如图 2 所示,并满足

$$x_{L2} - x_{L1} = P_L \left( 1 + \frac{g}{z} \right). \quad (9)$$

经过垂直相邻的两个透镜后,点  $A$  对应 EIP 上的  $I_1$  和  $I_2$  的  $y$  坐标满足

$$y_{L2} - y_{L1} = P_L \left( 1 + \frac{g}{z} \right). \quad (10)$$

光场中任意点  $A$  映射到每个 EI 上的像素点互称为同源点。由(9)式和(10)式可知,在 EI 阵列显示面上相邻同源点的间距相等,因此对于场景中每

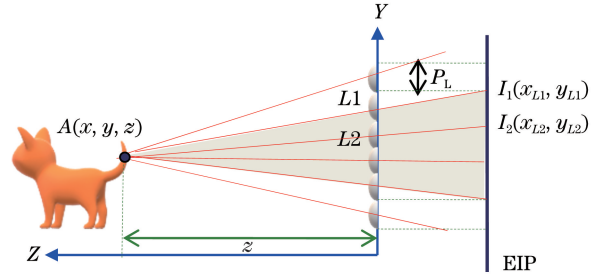


图 2 同一点在相邻两个 EI 上的坐标关系

Fig. 2 Coordinate relationship of same point on two adjacent EIs

个点,在水平和垂直两个方向映射到 EI 上的个数最多为  $P_L / \left[ P_L \left( 1 + \frac{g}{z} \right) - P_L \right] = z/g \leq z/g + 1$ ,所以最多可以求解  $(z/g + 1)^2$  个与 EI 对应的同源点。从图 2 可以看到,对于致密排列的透镜组,每个 EI 的显示区边长与透镜的直径是相等的。图中  $A$  点在垂直方向只有 4 个可成像的同源点,其他透镜对应的成像位置不在 EI 显示区内。

LF-EI 算法的计算过程简单且被普遍使用,但生成效率和质量低,主要存在如下缺陷。

1) 真实光场中深度发生阶跃变化的连接面(线),因为其在 2D 图像上是被遮挡或重合的,所以在映射后表现为不连续点,造成 EI 产生空洞(黑斑)的现象。LF-EI 算法存在的问题如图 3 所示。从图 3 可以看到,点  $A$  实际是光场中的  $AB$  线段,设  $AB$  与拍摄光线一致,简单理解光线为与  $Z$  轴平行; $AB$  线段是具有深度阶跃的连续线段,将其映射到 EI 上, $I_A$  和  $I_B$  之间会出现一个不连续的遗漏点区域,这些区域会影响再现光场的显示质量。

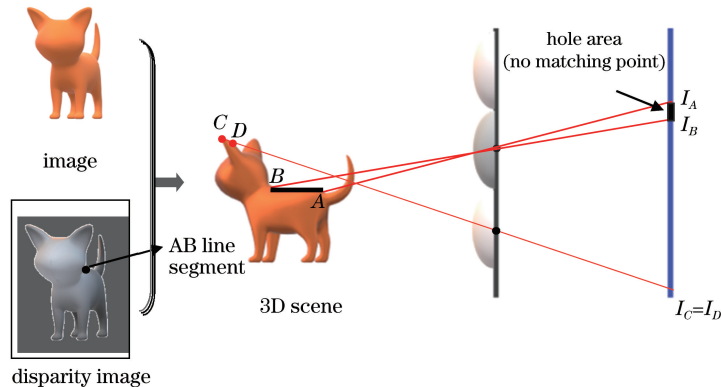


图 3 LF-EI 算法存在的问题

Fig. 3 Problem with LF-EI algorithm

2) 根据光场与 EIP 的映射关系,生成的 EI 是复杂的、多对一的冗余映射。

原始场景的几何尺寸是透镜的数倍以上, EI 与

透镜是等尺寸的,所以场景通过虚拟透镜投射到采集平面的像素数量是 EI 的数倍,且存在大量的冗余映射。冗余映射造成的主要影响:其一,需要计算场

景中每个点的映像,场景像素的总和决定运算时间,这一特点是导致该算法速率变慢的最主要因素之一;其二,EI在光场中沿光线方向倾斜的位置会出现多点(线)重复堆叠,导致显示信息覆盖,如图3 CD 线段;其三,EI使用整数坐标,所以对(7)~(10)式计算的映射点需要进行取整运算,导致某些相邻点重合。

#### 4 EIP 到 LF 的映射算法

针对 LF-EI 算法中的不足,提出一种快速生成 EI 的优化算法,简称 EI-LF 算法。EI-LF 算法生成 EI 的原理如图 4 所示。

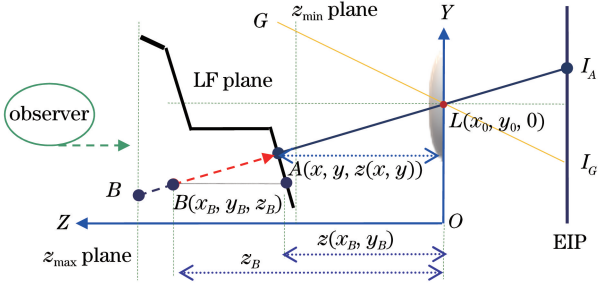


图 4 EI-LF 算法生成 EI 的原理

Fig. 4 Principle of EI-LF algorithm to generate EI

EI-LF 算法的核心:改变映射方向,建立从 EIP 到光场的映射函数,而不是从光场到 EIP。匹配原则:从 EIP 上的每一点出发,经过透镜光心和光场后与观察者之间作一条模拟的光路射线,沿着光路逆向搜索射线,距离观察者最近的光场上的点作为匹配点。

不同于 LF-EI 算法,EI-LF 算法是对 EI 上每一点直接求解其在光场中的唯一匹配点且没有冗余,计算时间只受控于 EI 阵列的像素和,而不是光场像素的总和,且基本不受光场分辨率的影响。因为每个 EI 的分辨率是投射场景的几十分之一或更小,所以极大地减少计算量。

图中  $z_{\min}$  和  $z_{\max}$  分别为光场的最小深度和最大深度所在的平面。为了计算 EI 上的点  $I_A(x_A, y_A)$ ,过透镜的光心  $L$  向光场方向作射线  $I_A L$ ,  $I_A L$  与光场曲面的交点  $A(x, y, z)$  作为  $I_A$  的匹配点。将(7)式和(8)式修改为

$$x = \left[ x_0 + (x_0 - x_A P_D) \frac{z}{g} \right] \frac{1}{P_1}, \quad (11)$$

$$y = \left[ y_0 + (y_0 - y_A P_D) \frac{z}{g} \right] \frac{1}{P_1}. \quad (12)$$

但(11)式和(12)式无法直接求解  $(x, y)$ ,因为  $(x, y)$  未求出之前,  $z$  是未知的。根据深度矩阵的

定义,只有光场表面上的点  $(x, y, z)$  满足  $z = z(x, y)$ ,光场曲面左侧的点  $(x, y, z)$  满足  $z > z(x, y)$ ,右侧的点  $(x, y, z)$  满足  $z < z(x, y)$ ,则交点  $A$  是光线  $I_A L$  上满足  $z = z(x, y)$  的唯一一点。根据这一特点可以使用 LF-EI 算法找到  $A(x, y, z)$ ,先求解  $I_A L$  与  $z_{\max}$  平面的交点  $B$ ,鉴于光场曲面是离散的像素,可以将点  $B$  沿光路反向并按像素步长向  $z_{\min}$  移动,迭代搜索满足  $z_B \leq z(x, y)$  的  $B$  点作为  $A$  点。具体方法如下。

首先,取  $z_B = z_{\max}$ ,并代入

$$x_B = \left[ x_0 + (x_0 - x_A P_D) \frac{z_B}{g} \right] \frac{1}{P_1}, \quad (13)$$

$$y_B = \left[ y_0 + (y_0 - y_A P_D) \frac{z_B}{g} \right] \frac{1}{P_1}, \quad (14)$$

得到  $x_B$  和  $y_B$ 。式中: $B(x_B, y_B, z_B)$  为  $I_A L$  与  $z_{\max}$  的交点。若恰好  $z_B = z(x_B, y_B)$ ,则  $A$  点就是  $B$  点。反之,  $z_B > z(x_B, y_B)$  或  $x_B, y_B$  超出光场范围。将  $B$  点沿  $BL$  线段并按步长  $P_1$  向  $L$  点移动,得到新的  $z_B$ ,将新的  $z_B$  代入(13)式和(14)式,进而算出新的  $(x_B, y_B)$ 。

重复移动点  $B$ ,直到  $z_B \leq z(x_B, y_B)$  或  $z_B < z_{\min}$ 。若  $z_B \leq z(x_B, y_B)$ ,这样求得的点  $B$  恰好是  $I_A L$  与光场曲面的交点  $A$ ,也就是  $I_A$  的匹配点,此时取  $I_A$  的像素值为  $A$  点像素值即可。若  $z_B < z_{\min}$ ,说明  $I_A L$  与光场无交点,  $I_A$  无匹配点,这种情况一般在 EI 的边缘区域可能出现,如图 4 中的光线  $I_G G$  所示。

当从  $z_{\max}$  平面向  $z_{\min}$  平面沿光线  $BL$  搜索  $I_A$  的匹配点  $A$  时,观察者位于光场的左侧,观察到的应该是在光线与光场的交点中距离观察者最近的点,即深度最大的交点。所以在沿光线方向重叠的位置,如图 5 中  $EF$  线段,直接选择光场中距离观察者最近的点  $E$ ,并将其作为映射的唯一匹配点,此时搜索停止。

计算过程中,对  $x_B, y_B$  进行取整处理。需要注意的是,为了保证对  $B$  点搜索的连续性,  $B$  点每次只能移动一个  $P_1$  长度,这样才能保证  $B$  点坐标的增量  $|\Delta x| \leq 1, |\Delta y| \leq 1$ 。

EI-LF 算法的优点如下。

1) 生成的 EI 质量更好,匹配点具有唯一性。

2) 因深度阶跃产生空洞的匹配问题是一个病态问题,仅依据单幅 2D 场景图像和深度信息无法知道阶跃区域侧面的形态,所以没有唯一解。EI-LF 映射是连续映射,可以对深度阶跃的区域给出更



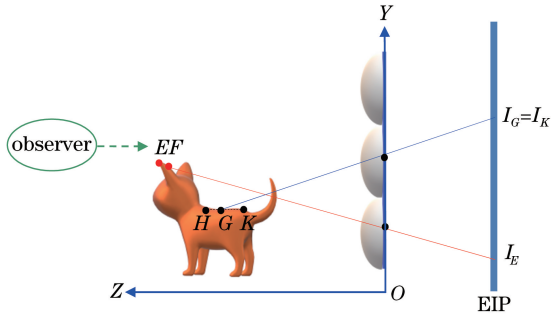


图 5 遮挡区域示意图

Fig. 5 Schematic of occluded area

合理的解决方案,避免 EI 上出现漏点(空洞)的现象。图 5 中  $HK$  按视线方向  $G$  点被  $H$  点遮挡,根据 EI-LF 算法,  $I_G$  的像素值与  $H$  点相等,即  $x_G = x_H, y_G = y_H, z_G < z(x_G, y_G) = z(x_H, y_H) = z_H, I_H = I_G = I_K$ ,这使得重现的光场有深度连续的视觉效果。

针对图 5 中  $HK$ ,全面的解决方案有 4 种:取距

表 1 两种算法的特点比较

Table 1 Comparison of characteristics of two algorithms

Algorithm	Structure	Matching process	Solution	Computational formula	Lens shape
LF-EI	Simpleness	Many-to-one	Map dead zone on EI	$S_{LF} \times S_{lens}$	Adapt
EI-LF	Complex	One-to-one	Four calculation methods	$S_{EI} \times S_{lens}$	Adapt

为了进一步提高算法的运算速率,EI-LF 算法可采取如下两项优化措施。

1) 当  $g > P_L/2$  时(对于绝大部分模型都是成立的),可以直接使用图 4 中  $B$  点深度坐标  $z_B$  进行迭代。迭代过程为  $z_{B,new} = z_{B,old} - P_1$ ,其中  $z_{B,old}$  为当前  $B$  点的深度坐标,  $z_{B,new}$  为移动后  $B$  点的深度坐标。使用每次计算的  $z_{B,new}$  替换(13)式和(14)式中的  $z_B$ ,进而解算出  $(x_B, y_B)$ ,直到  $z_B \leq z(x_B, y_B)$ ,这样不仅搜索步长更大,迭代次数更少,而且  $(x_B, y_B)$  的计算过程更简单。

2) 将距离的计量替换为 EIP 像素的间距  $P_D$ 。因为算法的最终目标是求出 EI 上  $I_A$  的像素值,与长度计量无关,这样(13)式和(14)式可以得到进一步简化。EI 显示区通常是正方形,设 EI 的分辨率为  $(2N+1) \times (2N+1)$ ,  $N$  为透镜半径,等于 EI 显示区边长的  $1/2$ 。对应透镜中心  $L(x_0, y_0)$  的 EI,其像素坐标取为  $(x_0+i, y_0+j)$ ,  $-N \leq i \leq N, -N \leq j \leq N$ 。此时,  $f$  和  $g$  均需按新的长度调整数值,  $P_L = N, g > N, x_0 - x_i = -i, y_0 - y_j = -j$ ,这可以显著降低单点计算的复杂度。

若使用  $P_D$  作为距离的计量,首先需要将(2)~

离观察者最近点的像素值( $H$  点);取远端的像素值( $K$  点);对  $HK$  两点的 RGB 三原色分别进行线性插值并再次合成;放弃匹配,保持空洞。使用 4 种方案产生集成成像光场,其正视点几乎是无差别的,但侧视点在  $HK$  上则因不同的方案而不同。

3) 无论场景的 2D 图像尺寸多大,EI-LF 算法的运算时间只受控于 EI 阵列的像素总数。若 LF-EI 算法的映射冗余比为  $k$ ,即光场中平均有  $k$  个点反向映射为 EI 上的一个点,则 EI-LF 算法的生成速率为 LF-EI 算法的  $k+1$  倍以上,此处省略判别比较的时间。映射的冗余比越大,EI-LF 算法的速率优势越明显。

4) EI-LF 算法是针对一个透镜提出的计算方法,与透镜阵列的形状和布局无关,但其可以适用于各种透镜阵列,如六边形和曲面阵列等。

表 1 为两种算法的特点对比,其中  $S_{LF}$  为光场像素的总数,  $S_{lens}$  为透镜的总数,  $S_{EI}$  为 EI 像素的总数。

(5)式进行调整,得到  $P_1 = d/g$  和  $\Delta d = 2d^2/gN$ ,则

$$z_{\max, \min} = \frac{fg}{g-f} \left[ 1 \pm \frac{f}{(g-f)N} \right]. \quad (15)$$

将(15)式代入(6)式,可以求出深度矩阵  $[z(x, y)]_{m \times n}$

根据(13)~(15)式给出优化后具体的计算流程,如图 6 所示。为了方便说明,在第  $k$  次迭代求出的  $B$  点坐标记作  $(x_k, y_k, z_k)$ 。停止迭代后,得到光场中与 EI 上点  $I_A(x_A, y_A)$  匹配的点  $A(x_k, y_k, z_k)$ 。

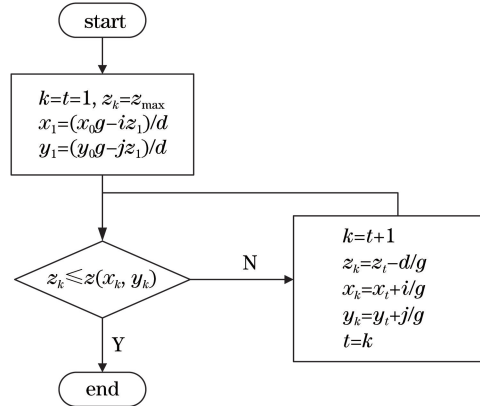


图 6 EI-LF 算法的流程图

Fig. 6 Flow chart of EI-LF algorithm

计算过程中还涉及范围检查和变量优化等处理,不在文中详述。

## 5 实验结果

### 5.1 不同算法的再现光场质量比较

为了检验所提算法的运行效果,使用数据库(<http://vision.middlebury.edu/stereo/data/>)提供的图像及视差图作为检测显示效果的单幅场景图像和深度信息来源,并进行如下实验。为了突显对照效果,对图像进行部分裁减,结果如图7所示。

透镜阵列由  $100 \times 100$  个紧密排列的方形孔径

微透镜构成,但实际仅使用  $63 \times 84$  个微透镜。每个微透镜的尺寸均为  $1 \text{ mm} \times 1 \text{ mm}$ ,  $f$  为  $3 \text{ mm}$ 。实验选用的其他参数如下, $g$  为  $3.5 \text{ mm}$ ,  $P_L = 1 \text{ mm}$ ,  $P_D = 0.05935 \text{ mm}$  ( $168.5 \text{ pixel/cm}$ )。由(1)~(3)式求出透镜到中心深度平面(CDP)的距离为  $21 \text{ mm}$ ,深度为  $14.955 \text{ mm}$ ,CDP 像素的直径  $P_1 = 0.356 \text{ mm}$ ,每个 EI 的分辨率为  $17 \text{ pixel} \times 17 \text{ pixel}$ 。使用图7生成的一个 EI 阵列,如图8所示。将 EI 阵列放置于立体成像显示平台上,并进行多视点拍摄,显示屏的分辨率为  $1920 \text{ pixel} \times 1080 \text{ pixel}$ ,实验效果如图9和图10所示。

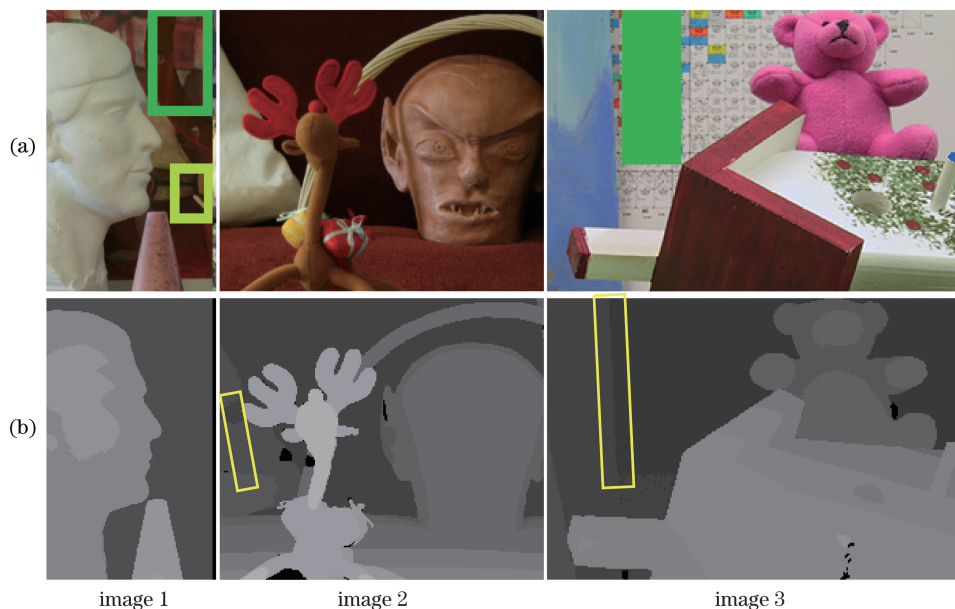


图7 测试图像。(a) 场景图像;(b) 视差图

Fig. 7 Test images. (a) Scene images; (b) disparity images

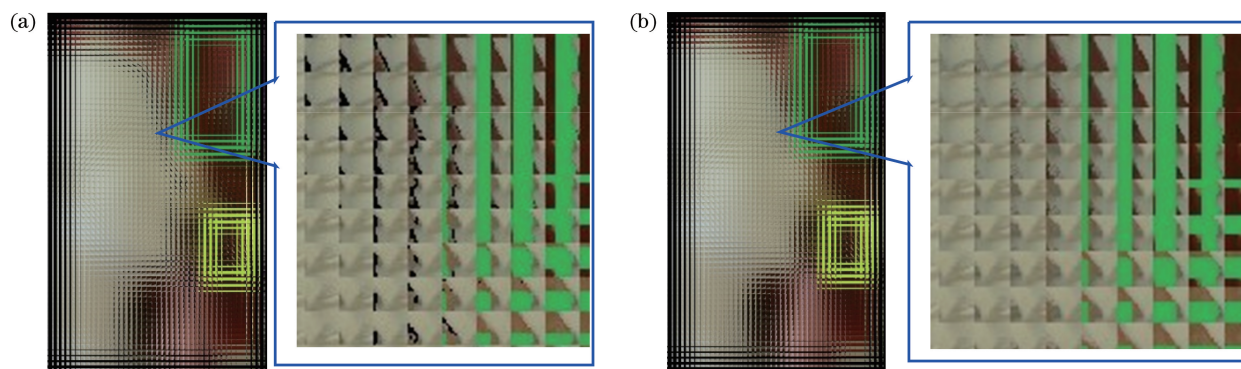


图8 两种算法生成的 EI 及局部放大图。(a) LF-EI 算法;(b) EI-LF 算法

Fig. 8 EI generated by two algorithms and partial enlarged images. (a) LF-EI algorithm; (b) EI-LF algorithm

从图10可以看到,不同视角间的视差明显,例如头像、鹿角和玩具房子的房顶均有明显移位。以上实验结果说明,所提的快速生成 EI 的算法是正确有效的。

通过对比可以发现,采用 EI-LF 算法重现的场景可以解决 LF-EI 算法存在空洞的问题,还原的光场更真实。采用 LF-EI 算法再现场景,因小熊区域等的深度相对背景变化较大,可以看到深度阶跃的



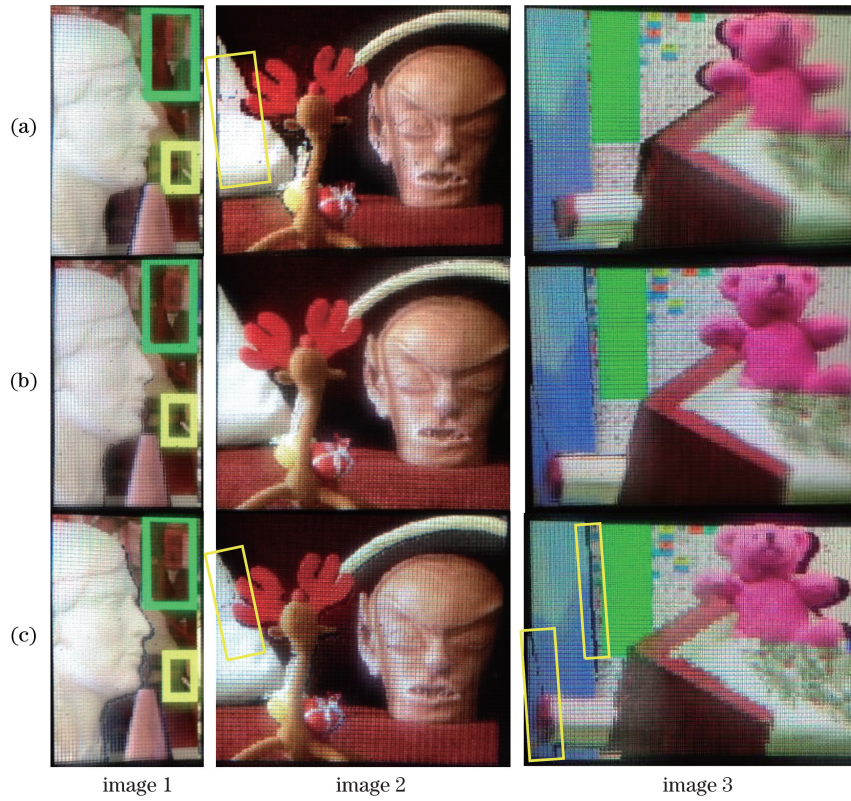


图 9 LF-EI 算法再现的 3D 光场多视点图像。(a)左视图;(b)正视图;(c)右视图

Fig. 9 3D light field multi-viewpoint images reproduced by LF-EI algorithm. (a) Left views; (b) front views; (c) right views

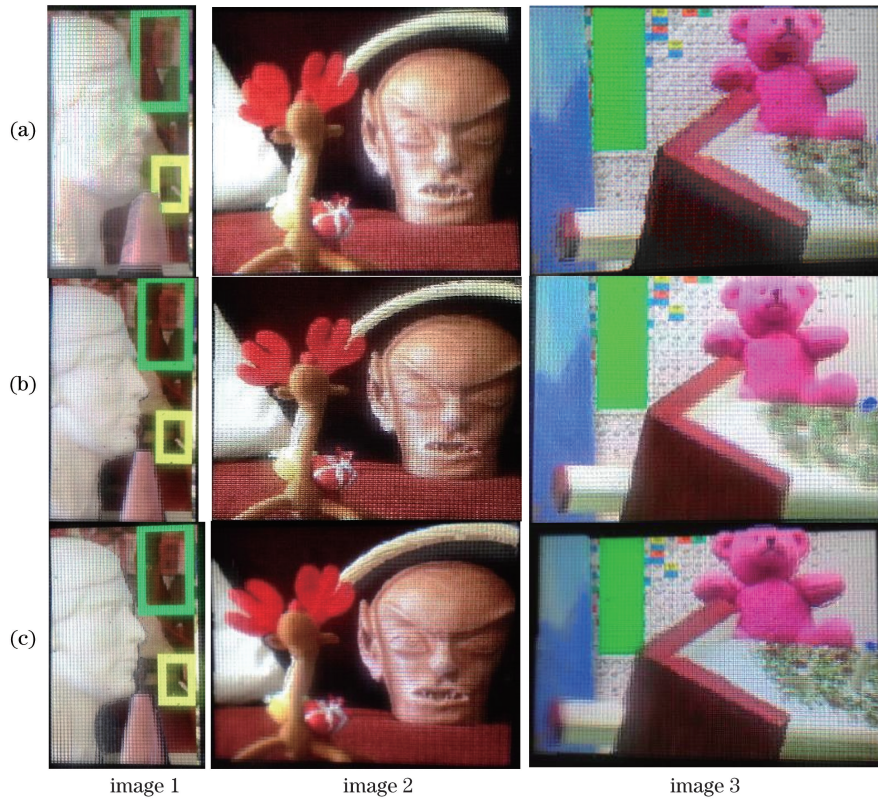


图 10 EI-LF 算法再现的 3D 光场多视点图像。(a)左视图;(b)正视图;(c)右视图

Fig. 10 3D light field multi-viewpoint images reproduced by EI-LF algorithm. (a) Left views; (b) front views; (c) right views

边界在某些视点上出现黑色空洞,如图 9 所示。虽然这种黑色空洞因错觉而表现出显著的立体感,但这种看法是错误的。图中下方方框标出的区域,因深度阶跃而出现黑色裂痕,但真实的场景应该是连续的曲面,不应该被撕裂。空洞的出现会降低还原光场的显示质量,而 EI-LF 算法则可以完全解决空洞的问题。

表 2 两种算法生成 EI 阵列的速率比较

Table 2 Comparison of rate of two algorithms to generate EI array

Serial No.	Scene image resolution / (pixel×pixel)	Sum of all pixels on scene image	Lens array (column×row)	EIA resolution / (pixel×pixel)	Sum of all pixels on EIP	Intel® Core™ i3-4010U CPU @ 1.70 GHZ /ms		Intel® Core™ i7-7500U @ 2.70 GHZ /ms	
						LF-EI	EI-LF	LF-EI	EI-LF
						1	353×241	85073	21×15
2	450×188	84600	33×16	562×273	153426	2167	288	1600	156
3	450×375	168750	33×28	562×477	268074	4337	506	3444	203
4	340×750	255000	26×50	443×851	376993	6568	728	5540	368
5	900×376	338400	59×28	1004×477	478908	9232	997	6933	442
6	678×750	508500	45×50	766×851	651866	13547	1335	10485	602
7	900×750	675000	60×50	1021×851	868871	17942	1692	14298	817

表 2 中数据均为单线程模式下采用缓冲区写屏的测试结果,未包含读取场景信息和深度信息的时间。使用图 8 作为测试图像,根据测试指定的分辨率对图像进行裁剪或放大。

从表 2 可以看到,LF-EI 算法的耗时与场景图像的像素总和接近正比例关系,而且对于不同尺寸的场景图像,LF-EI 算法的耗时均是 EI-LF 算法的 8 倍以上。

两种算法都支持多线程并行运算的模式。程序改用 4 线程并在 2.70 GHZ CPU 的笔记本计算机上进行计算,利用分辨率为 352 pixel×240 pixel 的场景图像和视差图生成 VCD(Video Compact Disc)标准的 EI 阵列。使用计数器累加生成帧数,LF-EI 算法的帧数只能达到 4.5 frame/s, EI-LF 算法为 43 frame/s。如果使用更快的硬件配置,如 8 核 CPU,数据采集和深度转换均使用独立线程来实现,通过 EI-LF 算法生成 EI 的速率完全可以支持 720 pixel×480 pixel 的 DVD(Digital Video Disc)标准视频。

为了保证生成图像的精准度和测试的可靠性,EI-LF 算法中的测试程序是根据每个透镜来独立计算 EI,并未使用(9)式和(10)式提供的同源点重用模式。这保证所提算法的测试结果可以适用于各种不同布局的微透镜阵列模型,同时也避免多点重叠。

## 5.2 不同算法的速率比较

针对 LF-EI 和 EI-LF 两种算法,根据第 4 节的最优方案,使用 Delphi 编程并在 PC 机上运行编译后的可执行程序,在不同配置的 PC(Personal Computer)机上使用不同分辨率的场景图像进行测试,结果如表 2 所示。

## 5.3 两种算法的时间复杂度分析

从原理上可以这样解释时间复杂度,从多到少的映射为  $f:U \rightarrow V$ ,若  $\forall x \in U$ ,求解  $y = f(x)$ ,效率肯定降低。改为反向,若  $\forall y \in V$ ,求解  $x = f^{-1}(y)$ ,且在两种映射复杂度相同的情况下,速度可以提高  $|U|/|V|$  倍,其中  $|U|$ 、 $|V|$  分别为  $U$ 、 $V$  的基数, $P$ 、 $Q$  分别为两个像素集合。

LF-EI 算法需要计算场景中各点映射到一个 EI 上的位置。同源点等距分布的特点能够简化计算过程,但由于映射是多选一的操作,需要在 EI 端进行预存储和比较,仍需要执行  $O(z_A/g + 1)^2$  次浮点运算,时间复杂度为  $O(n_0) + O\{n_0[(z_A/g) + 1]^2\}$ ,其中  $n_0$  为场景像素的总和, $z_A$  为场景各点深度的平均值。EI-LF 算法只需对 EI 的每个像素计算一次其在场景中的映射点即可,时间复杂度为  $O(n_1)$ ,其中  $n_1$  是 EI 阵列真正有图像区域的像素总和,无场景可匹配的用时极少,不予考虑。因此,LF-EI 算法的时间复杂度超过 EI-LF 算法数倍。

因为判断运算的用时极少,透镜数量远远小于像素数量,且 EI 阵列的像素总和已经包含透镜数量,所以分析两种算法的复杂度均不考虑优选判断环节和透镜数量的影响。

## 6 结 论

提出一种基于单幅彩色场景图像和深度信息快



速生成EI的EI-LF算法。该算法沿光路从EI的每个像素到场景中查找唯一的且是最佳的匹配点。相较于LF-EI算法,EI-LF算法具有消除空洞、避免多选的冗余计算和生成速度更快等优点。实验结果表明,使用4线程并行运算的EI-LF算法可以满足集成成像立体视频显示的要求。

EI-LF算法根据光路建立唯一性映射的关系,对于多角度深度信息的模型具有更广泛的适应性,比如增加被遮挡区域的空隙深度后,在算法不变的情况下,仍可以更好地应用分层的深度数据,沿光路找到穿过空隙遮挡区域的最佳匹配点。

### 参 考 文 献

- [1] Shen X, Javidi B. Large depth of focus dynamic micro integral imaging for optical see-through augmented reality display using a focus-tunable lens [J]. *Applied Optics*, 2018, 57(7): B184-B189.
- [2] Zhao X W, Wang Y Y, Song L P, et al. Underwater target imaging based on computational integral imaging [J]. *Chinese Journal of Lasers*, 2018, 45(1): 0109001. 赵雪微, 王尧尧, 宋丽培, 等. 基于计算集成成像的水下目标成像[J]. *中国激光*, 2018, 45(1): 0109001.
- [3] Wang J G, Xiao X, Hua H, et al. Augmented reality 3D displays with micro integral imaging [J]. *Journal of Display Technology*, 2015, 11(11): 889-893.
- [4] Xiao X, Javidi B, Martinez-Corral M, et al. Advances in three-dimensional integral imaging: sensing, display, and applications [J]. *Applied Optics*, 2013, 52(4): 546-560.
- [5] Xing Y, Wang Q H. 3D information acquisition technology of integral imaging [J]. *Infrared and Laser Engineering*, 2020, 49(3): 030300. 邢妍, 王琼华. 集成成像3D信息获取技术[J]. *红外与激光工程*, 2020, 49(3): 030300.
- [6] Li J J, Deng H, Li S, et al. One-step shooting method for integral imaging without depth inversion [J]. *Chinese Journal of Lasers*, 2020, 47(1): 0109002. 李姣姣, 邓欢, 李赛, 等. 无深度反转的集成成像一次拍摄方法[J]. *中国激光*, 2020, 47(1): 0109002.
- [7] Min S, Kim J, Lee B. New characteristic equation of three-dimensional integral imaging system and its applications [J]. *Japanese Journal of Applied Physics*, 2005, 44(2): 71-74.
- [8] Park K S, Min S W, Cho Y. View point vector rendering for efficient elemental image generation [J]. *IEICE Transactions on Information and Systems*, 2007, E90-D(1): 233-241.
- [9] Jang Y H, Chan P, Jung J S, et al. Integral imaging pickup method of bio-medical data using GPU and octree [J]. *The Journal of the Korea Contents Association*, 2010, 10(6): 1-9.
- [10] Xing S J, Sang X Z, Yu X B, et al. High-efficient computer-generated integral imaging based on the backward ray-tracing technique and optical reconstruction [J]. *Optics Express*, 2017, 25(1): 330-338.
- [11] Li G, Kwon K, Shin G, et al. Simplified integral imaging pickup method for real objects using a depth camera [J]. *Journal of The Optical Society of Korea*, 2012, 16(4): 381-385.
- [12] Zhao M, Xiong Z L, Xing Y, et al. Real-time integral imaging pickup system based on binocular stereo camera [J]. *Infrared and Laser Engineering*, 2017, 46(11): 1103007. 赵敏, 熊召龙, 邢妍, 等. 采用双目立体相机的实时集成成像拍摄系统[J]. *红外与激光工程*, 2017, 46(11): 1103007.
- [13] Hong S, Ansari A, Saavedra G, et al. Integral-imaging display from stereo-kinect capture [J]. *Proceedings of SPIE*, 2017, 1021: 102190K.
- [14] Deng H, Wang Q H, Li D H. Method of generating orthoscopic elemental image array from sparse camera array [J]. *Chinese Optics Letters*, 2012, 10(6): 061102.
- [15] Ouyang P, Yin S Y, Zhang Y C, et al. A fast integral image computing hardware architecture with high power and area efficiency [J]. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2015, 62(1): 75-79.
- [16] Wang Y, Yang J X, Liu L, et al. Computation reconstruction of integral imaging based on elemental images stitching [J]. *Acta Optica Sinica*, 2019, 39(11): 1110001. 王宇, 杨金晓, 刘乐, 等. 基于元素图像拼接的集成成像计算重构[J]. *光学学报*, 2019, 39(11): 1110001.
- [17] Pham D Q, Kim N, Kwon K C, et al. Depth enhancement of integral imaging by using polymer-dispersed liquid-crystal films and a dual-depth configuration [J]. *Optics Letters*, 2010, 35(18): 3135-3137.
- [18] Kim Y, Choi H, Kim J, et al. Depth-enhanced integral imaging display system with electrically variable image planes using polymer-dispersed liquid-crystal layers [J]. *Applied Optics*, 2007, 46(18): 3766-3773.