

基于窗截取的立体元图像阵列快速生成

王伟^{1,2}, 王世刚¹, 赵岩¹, 钟诚³

¹吉林大学通信工程学院, 吉林 长春 130012;

²长春工业大学计算机科学与工程学院, 吉林 长春 130012;

³长春工业大学公共管理学院(人文学院), 吉林 长春 130012

摘要 集成成像需要从不同角度记录三维(3D)物体的空间信息,采用计算机生成时,计算量大、时间长。针对这一问题,提出窗截取的立体元图像阵列快速生成方法。模拟真实透镜阵列的结构,建立采样模型,根据显示平台光学参数计算得出虚拟 3D 物体对应每个虚拟透镜元中的图像,即立体元图像,然后采用窗截取的方式生成立体元图像阵列。改变采样点和窗函数可以生成任意孔径任意排列结构的立体元图像阵列。实验搭建基于 LED 的集成成像显示平台,设计与 LED 匹配的方形、六边形、圆形孔径的透镜阵列,选取不同类型的 3D 模型对比立体元图像阵列的计算时间和立体显示效果,结果表明,在不改变立体图像质量的前提下,当立体元图像的分辨率高于透镜阵列的采样率时,本文方法速度更快。

关键词 图像处理; 集成成像; 透镜阵列; 立体元图像

中图分类号 TN911.73

文献标识码 A

doi: 10.3788/AOS201838.0710002

Fast Elemental Image Array Generation Based on Window Interception

Wu Wei^{1,2}, Wang Shigang¹, Zhao Yan¹, Zhong Cheng³

¹College of Communication Engineering, Jilin University, Changchun, Jilin 130012, China;

²School of Computer Science & Engineering, Changchun University of Technology, Changchun, Jilin 130012, China;

³School of Public Administration (School of Humanities), Changchun University of Technology, Changchun, Jilin 130012, China

Abstract Integral imaging needs to record the spatial information of three-dimensional (3D) objects from different perspectives, therefore the computation is large and cost a long time by computer-generated method. To solve this problem, we present a fast elemental image array generation method by using window interception. The sampling model is established by simulating the structure of the lenslet array, and the elemental images of the virtual 3D object, corresponding to each virtual lens, are calculated based on optics parameters of the display platform. Then, the window interception is used to generate the elemental image array. The elemental image array with any aperture and any arranging structure can be generated by changing the sampling points and the window function. We build an integral imaging display platform based on LED and design the lenslet arrays with the shape of square, hexagonal and circular aperture. Different 3D models are used to compare the processing time and display effect. Experimental results show that without reducing the 3D image quality, the proposed method is faster when the resolution of elemental image is higher than the sampling rate of the lenslet array.

Key words image processing; integral imaging; lenslet array; elemental image

OCIS codes 100.6890; 100.6880; 100.2000

收稿日期: 2018-02-05; 收到修改稿日期: 2018-03-07

基金项目: 国家自然科学基金重点项目(61631009)、国家十三五重点研发计划项目(2017YFB0404800)、中央高校基本科研业务费专项资金(2017TD-19)

作者简介: 王伟(1982—),女,博士研究生,主要从事立体视频处理方面的研究。E-mail: 114188246@qq.com

导师简介: 王世刚(1962—),男,博士,教授,博士生导师,主要从事立体视频处理方面的研究。

E-mail: wangshigang@vip.sina.com

* 通信联系人。E-mail: wangshigang@vip.sina.com

1 引 言

自 1908 年集成成像技术被首次提出以来,很多立体元图像阵列的生成方法也随之产生^[1-6]。计算性集成成像一直是学者研究的热点^[7]。目前,计算性集成成像主要有平行组渲染、视点向量渲染、多视点渲染、单视点独立渲染等。平行组渲染利用聚焦模式的显示特性,将每一个透镜元作为一个立体像素点,因此只适用于聚焦显示模式^[8];视点向量渲染比平行组渲染简单,但是存在三维(3D)图像失真的问题^[9]。多视点渲染模拟真实透镜阵列的采集过程,对场景、显示模式没有限制,能较好地还原 3D 物体的几何位置、色彩、纹理等^[10]。然而受 3D 物体尺寸、色彩、纹理的影响,通常渲染时间较长^[11-15]。平行组、视点向量、多视点渲染可以看作是为了加速生成单个视点图像的特殊单视点独立渲染。作者在以往的计算性集成成像研究中发现^[16],虽然这些方法不同程度地降低了渲染时间,但当立体元图像的分辨率增加时,渲染时间均会明显增加^[11,17]。这是由于传统的单视点独立渲染要生成多种孔径透镜阵列所需的立体元图像阵列,首先要生成矩形的视点图像,然后利用反向光线追踪计算视点图像中每个像素点在对应立体元图像阵列中的位置,虚拟相机的数量与立体元图像的分辨率一致。目前,随着高清产品的不断推出,显示设备的分辨率可达 736 pixel/inch

(1 inch=2.54 cm),而透镜阵列受制作工艺的限制,孔径一般为 1~10 mm,因此很多集成成像系统的立体元图像分辨率均高于透镜阵列的采样率。虚拟相机数量是影响渲染时间的主要原因。

为此,本文提出基于窗截取的立体元图像阵列快速生成法。根据透镜阵列采样结构建立虚拟相机阵列,采用平行拍摄的方式,直接获取 3D 物体对应的立体元图像,然后利用窗截取并拼接生成不同透镜阵列所需的立体元图像阵列。改变采样结构和窗函数可以生成任意孔径任意排列结构的立体元图像阵列。搭建基于 LED 的集成成像显示平台,并设计与之匹配的方形、圆形、六边形透镜阵列。当立体元图像的分辨率高于透镜阵列的采样率时,该方法简单快速。

2 基本原理

本文方法的基本原理如图 1 所示。根据真实透镜阵列结构建立采样模型,由显示平台光学参数计算得出虚拟 3D 物体对应每个虚拟透镜元中的图像,即立体元图像。在计算中,无须考虑真实透镜的有限孔径引起的衍射和离焦引起的模糊。因此,计算生成的视点图像就是 3D 物体在采集平面的投影。利用窗函数将计算的视点图像根据虚拟透镜元的形状进行裁剪,最后通过拼接的方式,生成对应的立体元图像阵列。

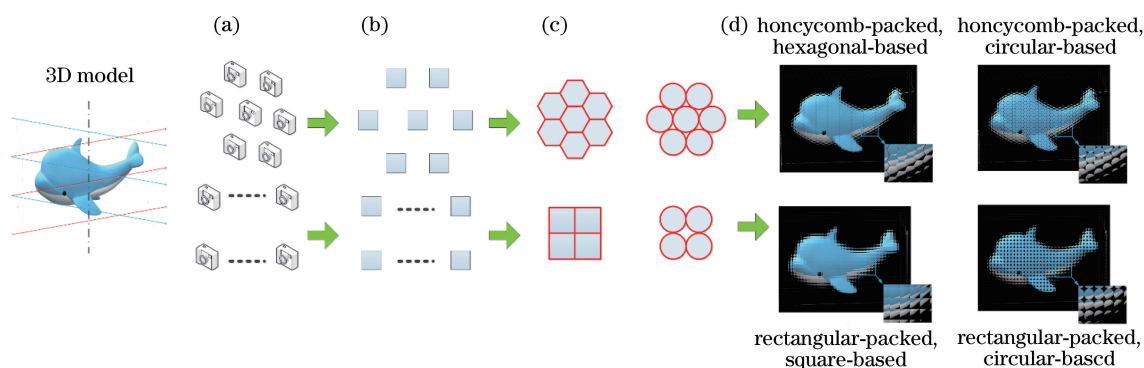


图 1 立体元图像阵列的生成。(a)虚拟相机阵列;(b)立体元图像;(c)窗截取与拼接;(d)立体元图像阵列

Fig. 1 Generation of elemental image array. (a) Virtual camera array; (b) elemental image; (c) window interception and image mosaic; (d) elemental image array

3 窗截取的立体元图像阵列快速生成

本文方法可以生成任意孔径透镜阵列所需的立体元图像阵列。目前常用的透镜阵列为圆形、方形、六边形孔径透镜阵列^[18],如图 2 所示,以这三种孔径为例,阐述窗截取的任意立体元图像阵列的快速

生成方法。

3.1 采样模型

圆形、方形、六边形孔径透镜阵列,采样结构可以归为矩形采样和蜂窝采样两种^[19]。设每个透镜元的孔径均为 P ,则透镜阵列的采样结构如图 3 所示。

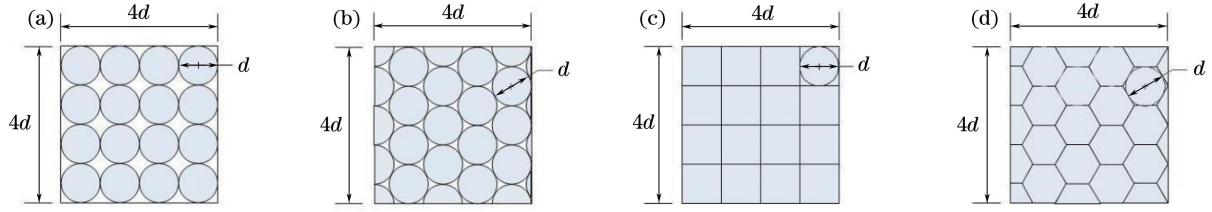


图 2 内切圆直径均为 d 的透镜阵列结构。(a)圆形孔径矩阵排列;(b)圆形孔径蜂窝排列;
(c)方形孔径矩阵排列;(d)六边形孔径蜂窝排列

Fig. 2 Structure of lenslet arrays with the same inscribe circle diameter of d . (a) Rectangular-packed, circular-based array;
(b) honeycomb-packed, circular-based array; (c) rectangular-packed, square-based array;
(d) honeycomb-packed, hexagonal-based array

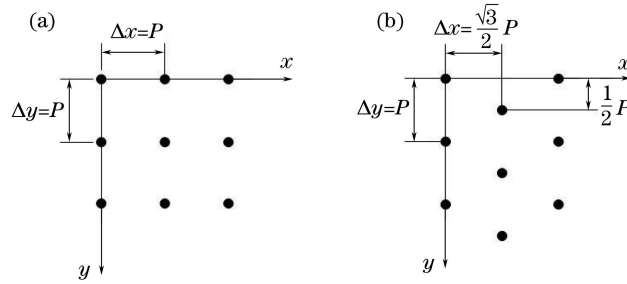


图 3 (a)矩形采样与(b)蜂窝采样示意图

Fig. 3 Diagram of (a) rectangular-packed sampling and (b) honeycomb-packed sampling

设左上角虚拟透镜元中心为原点,则透镜阵列中每个透镜元的空间采样点坐标 S_r 、 S_h 分别为

$$S_r(x, y) = [(x - 1)P, (y - 1)P],$$

$$x, y = 1, 2, 3, \dots, \quad (1)$$

$$S_h(x, y) =$$

$$\begin{cases} [(x - 1)P, (y - 1) \frac{\sqrt{3}}{2}P], \\ x = 1, 2, 3, \dots, y = 1, 3, 5, \dots \end{cases}, \quad (2)$$

$$\begin{cases} [\frac{P}{2} + (x - 1)P, \frac{\sqrt{3}}{2}P + (y - 2) \frac{\sqrt{3}}{2}P], \\ x = 1, 2, 3, \dots, y = 2, 4, 6, \dots \end{cases}$$

式中下标 r 代表矩形采样, h 代表蜂窝采样。

3.2 参数设计

每个透镜元的视角 θ 、所含像素数量 Q , 以及立体图像的景深 D 要根据显示平台中所使用的透镜阵列的光学参数及显示屏参数设置, 设透镜元焦距为 f , 显示屏的分辨率为 R_d (单位为 $\text{pixel} \cdot \text{mm}^{-1}$), 则每个透镜元的视角为

$$\Omega_s = 2\arctan\left(\frac{P}{2f_s}\right), \quad (3)$$

$$\begin{cases} \Omega_{hh} = 2\arctan\left(\frac{P}{\sqrt{3}f_h}\right) \\ \Omega_{hv} = 2\arctan\left(\frac{P}{2f_h}\right) \end{cases}. \quad (4)$$

所含像素数量为

$$Q_s = PR_d, \quad (5)$$

$$Q_{hh} = \frac{2\sqrt{3}}{3}PR_d, \quad Q_{hv} = PR_d. \quad (6)$$

立体图像的深度为

$$D \approx Qf, \quad (7)$$

式中下标 s 代表方形和圆形孔径透镜阵列, hh、hv 分别代表六边形孔径透镜阵列的水平 and 垂直方向。

3.3 窗截取并拼接

利用窗函数将每个虚拟相机采集的立体元图像进行裁剪, 并根据透镜阵列采样结构将裁剪后的立体元图像拼接成立体元图像阵列。若生成方形孔径透镜阵列所需的立体元图像阵列, 直接将立体元图像以矩形采样的方式拼接即可。若生成圆形和六边形孔径透镜所需的立体元图像阵列, 则采用圆形和六边形窗函数, 将立体元图像裁剪, 并按矩形或蜂窝式采样结构拼接。如图 4(a) 所示, 设立体元图像的左上角为坐标原点, 分辨率为 $Q_s \times Q_s$ 、 $Q_{hh} \times Q_{hv}$, 窗函数的中心与立体元图像的中心一致, 则圆形窗函数的半径为 $R = \frac{Q_s}{2}$, 六边形窗函数在每个立体元图像中的顶点坐标为

$$\begin{cases} A = \left(0, \frac{1}{4}Q_{hh}\right), & B = \left(0, \frac{3}{4}Q_{hh}\right) \\ C = \left(\frac{1}{2}Q_{hv}, Q_{hh}\right), & D = \left(Q_{hv}, \frac{3}{4}Q_{hh}\right), \\ E = \left(Q_{hv}, \frac{1}{4}Q_{hh}\right), & F = \left(\frac{1}{2}Q_{hv}, 0\right) \end{cases} \quad (8)$$

如图 4(b)所示,若虚拟透镜阵列由 $m \times n$ 个透镜组成,则第 (i, j) 个立体元图像 $E(x, y)$ 在矩阵采样的立体元图像阵列 $R(X, Y)$ 中的拼接点坐标为

$$E(x, y) = R[X + Q_s(i - 1), Y + Q_s(j - 1)], \quad i, j = 1, 2, 3, \dots, \quad (9)$$

式中立体元图像阵列的分辨率为 $mQ_s \times nQ_s$ 。

$E(x, y)$ 在蜂窝采样的立体元图像阵列 $H(X, Y)$ 中的拼接点坐标为

$$E(x, y) = \begin{cases} H\left[X + Q_{hv}(i - 1), Y + \frac{3}{4}Q_{hh}(j - 1)\right], \\ i = 1, 2, 3, \dots, j = 1, 3, 5, \dots \\ H\left[X + \frac{1}{2}Q_{hv} + Q_{hv}(i - 1), Y + \frac{3}{4}Q_{hh} + \frac{3}{4}Q_{hh}(j - 2)\right], \\ i = 1, 2, 3, \dots, j = 2, 4, 6, \dots \end{cases} \quad (10)$$

式中立体元图像阵列的分辨率为

$$\left(m \times Q_{hv} + \frac{1}{2} \times Q_{hv}\right) \times \left[Q_{hh} + (n - 1) \times \frac{3}{4}Q_{hh}\right]。$$

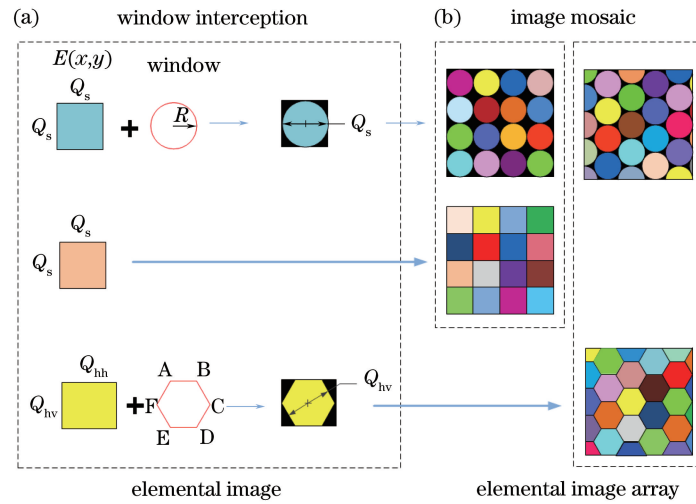


图 4 (a)窗截取与(b)图像拼接

Fig. 4 (a) Window interception and (b) image mosaic

3.4 算法比较

图 5(a)和图 5(b)所示分别为本文方法和单视点独立渲染^[11,17]的过程。单视点独立渲染均为矩阵采样,采用会聚光线追踪渲染,首先生成的是子图像(视点图像),然后根据透镜阵列结构,利用反向光线追踪,逐点计算子图像中每个像素点在不同立体元图像阵列中的坐标,虚拟相机的数量与立体元图像的分辨率一致。本文方法根据透镜阵列结构建立采样模型,采用平行光线追踪渲染,直接生成立体元图像,利用窗截取将立体元图像裁剪并拼接,生成立体元图像阵列,虚拟相机的数量与透镜阵列采样率(所含透镜元数量)一致。当立体元图像的分辨率大于透镜元的数量时,本文方法的虚拟相机数量少于单视点独立渲染,因此渲染时间更短。综上,本文方法更适用于立体元图像分辨高于透镜元数量的情

况。平行与会聚渲染的区别如图 5(c)所示,子图像与立体元图像的关系如图 5(d)所示。

4 实验结果

搭建基于发光二极管(LED)的集成成像显示平台,设计与 LED 匹配的方形、六边形、圆形孔径的透镜阵列,如图 6 所示。通过导轨调节透镜阵列,使 LED 显示的立体元图像与透镜元对齐。

将本文方法与文献[17]方法进行对比,首先对比两种方法的计算时间,然后采用主观和客观两种方式比较重建立体图像的质量。由于计算时间受 3D 模型的色彩、纹理及立体元图像分辨率等影响,故选用简单字母“JLU”和复杂纹理的“海豚”作为 3D 模型。如图 7(a)所示。图 7(b)~(d)为两种方法生成的圆形、方形和六边形孔径的立体元图像阵

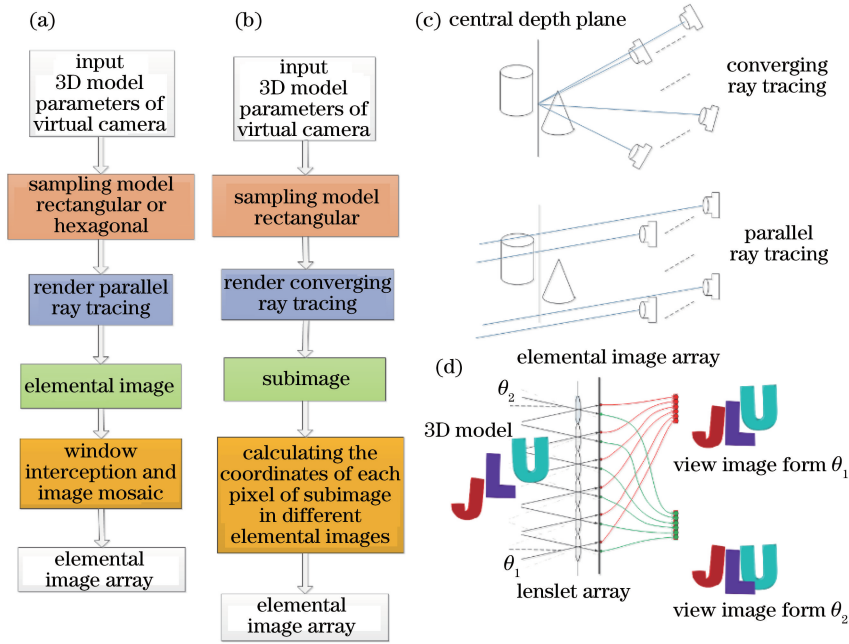


图 5 不同算法比较。(a)本文方法;(b)单视点独立渲染;(c)不同光线追踪渲染法;(d)子图像与立体元图像的对应关系
Fig. 5 Comparison of different algorithms. (a) Proposed method; (b) single viewpoint independent rendering; (c) different ray tracing rendering methods; (d) relation between subimage and elemental image

列。海豚模型对应本文方法,字母“JLU”对应文献[17]方法,3D模型与显示平台参数如表1所示。LED的分辨率为 $384\text{ pixel}\times 384\text{ pixel}$,像素间距为 1.25 mm 。硬件配置为Intel(R) Core(TM) i7-7700 CPU @ 4.2 GHz, 16 G RAM NVIDIA GTX 1070显卡。

当透镜元数量、立体元图像分辨率和3D模型均相同时,不同形状立体元图像阵列的计算时间近似相等。实验对方形立体元图像阵列在不同透镜元数量、立体元图像分辨率和3D模型下的计算时间进行比较。在透镜阵列为 $24\text{ pixel}\times 24\text{ pixel}$ 、立体元图像阵列分辨率为 $70\text{ pixel}\times 70\text{ pixel}$ 时,本文方法对JLU和海豚模型的计算时间分别为 18.47 s 和 33.63 s ;文献[17]方法对JLU和海豚模型的计算时

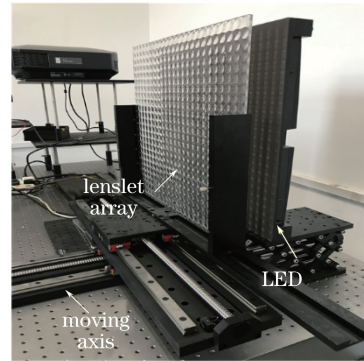


图 6 LED 集成成像显示平台

间分别为 640.71 s 和 749.28 s 。可以看出,针对不同的3D模型,本文方法的计算时间均少于文献[17]方法。

表 1 3D模型与显示平台参数

Table 1 3D models and display platform parameters

3D model	Top	Face	Triangle
JLU	2640	2406	5268
Dolphin	11302	11296	22592
Lenslet array	Array size /(pixel×pixel)	Lens pitch /mm	Focal length /mm
Square	48×48	10	60
Hexagon	48×55	10	60
Circular	24×24	20	60

图8所示为本文方法和文献[17]方法在透镜阵列为 $24\text{ pixel}\times 24\text{ pixel}$ 、立体元图像分辨率不同的情况下,JLU和海豚模型的计算时间比较。

从图8可以看出,针对不同的3D模型,随着立体元图像分辨率的增加,文献[17]方法的计算时间明显增加。而本文方法受立体元图像分辨的影响很

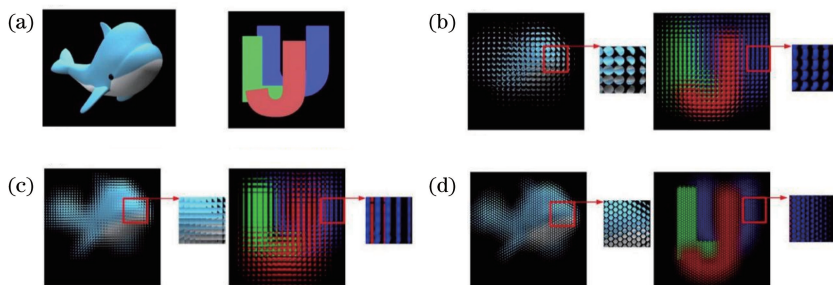


图 7 (a) 3D 模型;(b)圆形立体元图像阵列;(c)方形立体元图像阵列;(d)六边形立体元图像阵列
Fig. 7 (a) 3D models; (b) circular elemental image arrays; (c) square elemental image arrays; (d) hexagonal elemental image arrays

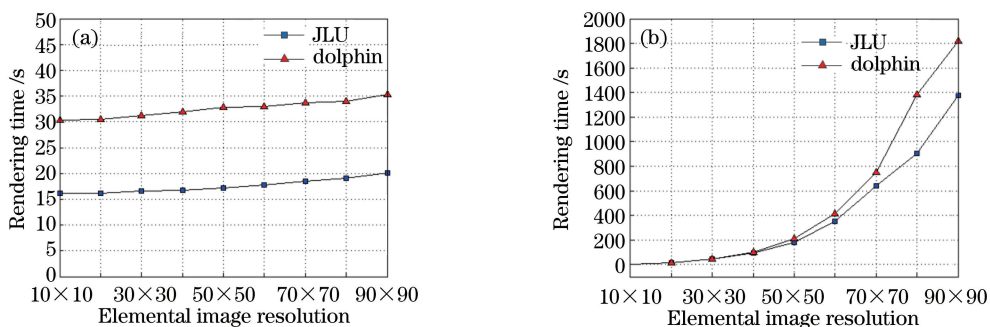


图 8 透镜阵列 24 pixel×24 pixel 时,(a)本文方法和(b)文献[17]方法在 JLU 和海豚模型上的计算时间比较
Fig. 8 Comparison of rendering time between (a) proposed method and (b) method in Ref. [17] on JLU and dolphin, when the lenslet array is 24 pixel×24 pixel

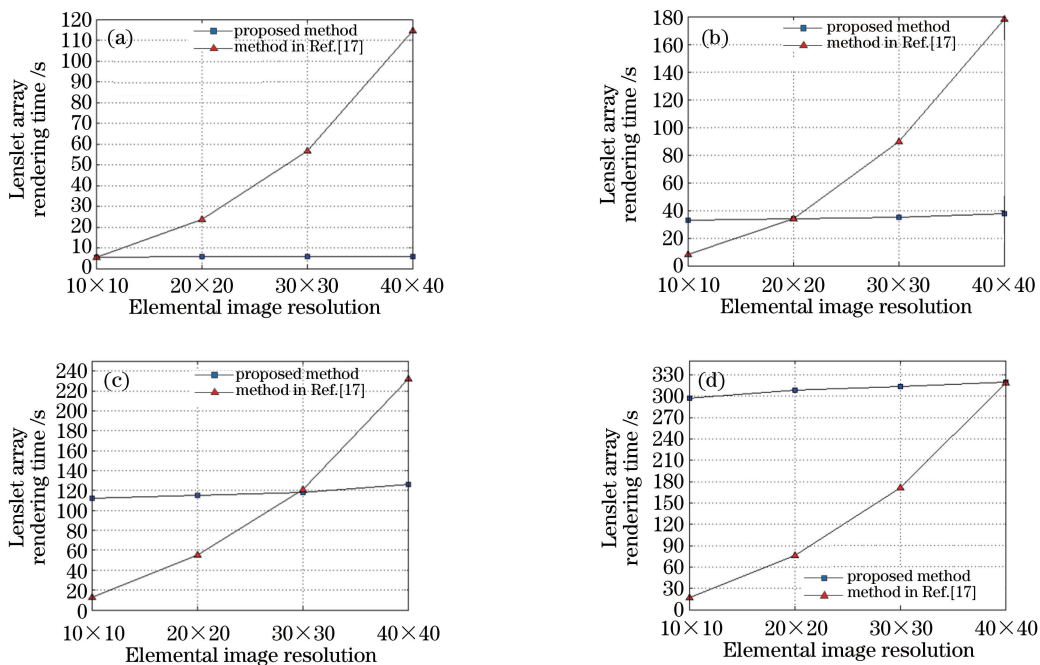


图 9 本文方法与文献[17]方法在不同透镜阵列下的计算时间比较。

(a) 10 pixel×10 pixel;(b) 20 pixel×20 pixel;(c) 30 pixel×30 pixel;(d) 40 pixel×40 pixel

Fig. 9 Comparison of processing time between the proposed method and method in Ref. [17] under different lenslet arrays.

(a) 10 pixel×10 pixel; (b) 20 pixel×20 pixel; (c) 30 pixel×30 pixel; (d) 40 pixel×40 pixel

小,计算时间更短。

图 9 所示为本文方法和文献[17]方法在透镜阵列和立体元图像分辨率均不同的情况下,海豚模型计算时间的比较。可以看出,当立体元图像分辨率高于透镜元的数量时,本文方法计算用时更短。

此外,本文方法在减少计算时间的同时,并未降低立体图像质量。利用本文方法和文献[17]方法分

别生成圆形、方形和六边形立体元图像阵列,对比计算机重建和光学重建的立体图像质量。图 10 为计算机对方形和六边形立体元图像阵列所生成的立体图像的重建。比较不同深度重建图像与参考图像[图 7(a)海豚]的峰值信噪比(PSNR),如图 11 所示。图 12 为方形和圆形立体元图像阵列的光学重建。

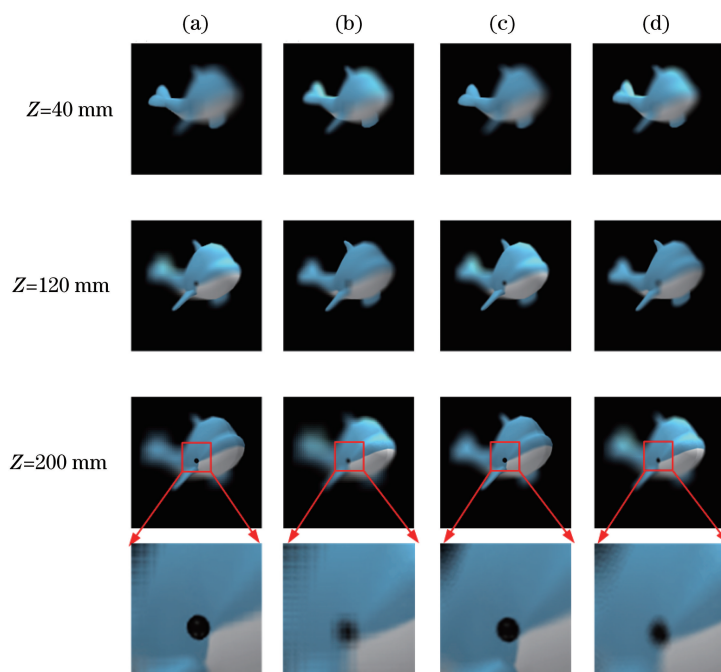


图 10 不同深度平面立体图像的计算机重建。(a)(c)本文方法;(b)(d)文献[17]方法;(a)(b)方形立体元图像阵列;(c)(d)六边形立体元图像阵列

Fig. 10 Computer reconstruction of stereo images at different depth planes. (a)(c) Proposed method; (b)(d) method in Ref. [17]; (a)(b) square elemental image array; (c)(d) hexagonal elemental image array

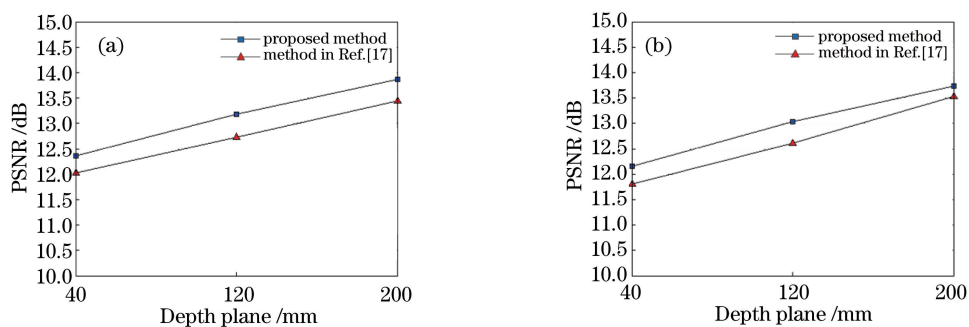


图 11 利用(a)方形和(b)六边形立体元图像阵列重建的不同深度平面立体图像的 PSNR 比较
Fig. 11 PSNR comparison between stereo images reconstructed at different depth planes by (a) square elemental and (b) hexagonal elemental image array

分别对海豚的尾、鳍和眼进行计算机重建,如图 10 所示。可以看出,不同深度平面下,本文方法生成的立体图像质量均高于文献[17]方法。通过对不同深度平面立体图像的 PSNR 的比较,量化分析了重建立体图像的质量,如图 11。同样,在

不同深度平面,本文方法的结果均优于文献[17]方法。

图 12 所示为不同视角观看到的立体显示效果。可以看出,本文方法与文献[17]方法均可看到不同角度的海豚和字母 JLU 的不同遮挡情况。综上,主

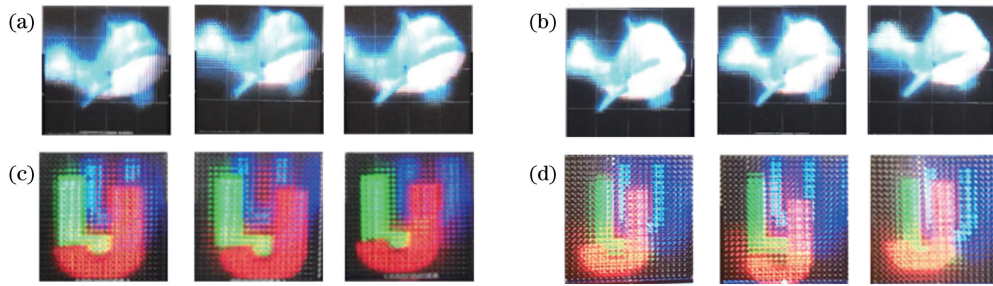


图 12 (a)(c)本文方法和(b)(d)文献[17]方法生成的(a)(b)方形、(c)(d)圆形立体元图像阵列在集成成像 LED 显示平台下的立体显示效果

Fig. 12 Stereoscopic display of LED-based integral imaging using (a)(b) square and (c)(d)circular elemental image arrays generated by (a)(c) proposed method and (b)(d) method in Ref. [17]

观与客观评价结果均说明,本文方法在加快渲染速度的同时,并没有降低立体图像的质量。

5 结 论

提出一种窗截取的立体元图像阵列快速生成方法,搭建基于 LED 的集成成像显示平台,设计与之匹配的方形、六边形、圆形孔径的透镜阵列,通过对比不同 3D 模型、不同分辨率的立体元图像的计算时间和真实立体显示效果,可以看出,在不改变立体图像质量的前提下,本文方法受立体元图像阵列分辨率的影响较小,当立体元图像的分辨率高于透镜阵列的采样率时,本文方法更简单快速,可以真实再现 3D 物体的空间信息。改变采样点和窗函数可以生成任意孔径任意排列结构的立体元图像阵列,提供集成成像不同显示平台所需的各种图像来源。

参 考 文 献

- [1] Lippmann M. Epreuves reversibles donnant la sensation du relief[J]. *Journal of Physiology*, 1908, 1(7): 821-825.
- [2] Ives H E. Optical properties of a Lippmann lenticulated sheet[J]. *Journal of the Optical Society of America*, 1931, 21(3): 171-176.
- [3] Hong J, Kim Y, Choi H J, *et al.* Three-dimensional display technologies of recent interest: principles, status, and issues[J]. *Applied Optics*, 2011, 50(34): H87-H115.
- [4] Arai J, Okano F, Hoshino H, *et al.* Gradient-index lens-array method based on real-time integral photography for three-dimensional images [J]. *Applied Optics*, 1998, 37(11): 2034-2045.
- [5] Javidi B, Sola-Pikabea J, Martinez-Corral M. Breakthroughs in photonics 2014: recent advances in 3-D integral imaging sensing and display[J]. *IEEE Photonics Journal*, 2015, 7(3): 0700907.
- [6] Igarashi Y, Murata H, Ueda M. 3-D display system using a computer generated integral photograph[J]. *Japanese Journal of Applied Physics*, 1978, 17(9): 1683-1684.
- [7] Park S G, Yeom J, Chen N, *et al.* Recent issues on integral imaging and its applications[J]. *Journal of Information Display*, 2014, 15(1): 37-46.
- [8] Min S W, Kim J, Lee B. New characteristic equation of three-dimensional integral imaging system and its applications[J]. *Japanese Journal of Applied Physics*, 2005, 44(1-7): 71-74.
- [9] Park K S, Min S W, Cho Y. Viewpoint vector rendering for efficient elemental image generation[J]. *IEICE Transactions on Information and Systems*. 2007, 90(1): 233-241.
- [10] Halle M. Multiple viewpoint rendering [C]. *Conference on Computer Graphics and Interactive Techniques*, 2010: 243-254.
- [11] Xing S, Sang X, Yu X, *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.
- [12] Li S L, Wang Q H, Xiong Z L, *et al.* Multiple orthographic frustum combing for real-time computer-generated integral imaging system [J]. *Journal of Display Technology*, 2014, 10(8): 704-709.
- [13] Kim J, Jung J H, Jang C, *et al.* Real-time capturing and 3D visualization method based on integral imaging [J]. *Optics Express*, 2013, 21(16): 18742-18753.
- [14] Kwon K C, Park C, Erdenebat M U, *et al.* High speed image space parallel processing for computer-generated integral imaging system [J]. *Optics Express*, 2012, 20(2): 732-740.
- [15] Kang H H, Shin D H, Kim E S. Efficient compression of motion-compensated sub-images with Karhunen-Loeve transform in three-dimensional

- integral imaging[J]. Optics Communications, 2010, 283(6): 920-928.
- [16] Wu W, Wang S G, Wang H Z, *et al.* Elemental image generation based on Maya[J]. Journal of Jilin University (Engineering and Technology Edition), 2017, 47(4): 1314-1320.
- 武伟, 王世刚, 王宏志, 等. 基于 Maya 的立体元图像阵列的生成[J]. 吉林大学学报(工学版), 2017, 47(4): 1314-1320.
- [17] Yanaka K. Integral photography using hexagonal fly's eye lens and fraction view[J]. SPIE, 2008, 6803: 68031K.
- [18] Liu D S. Micro optics and microlens arrays [M]. Beijing: Science Press, 2013.
- 刘德森. 微小光学与微透镜阵列[M]. 北京: 科学出版社, 2013.
- [19] Chen N, Yeom J, Jung J H, *et al.* Resolution comparison between integral-imaging-based hologram synthesis methods using rectangular and hexagonal lens arrays [J]. Optics Express, 2011, 19 (27): 26917-26927.