

文章编号:1002-2082 (2020) 01-0086-08

基于 FPGA 的图像畸变矫正算法研究

毛昕蓉, 刘铠铭, 汪乐意, 韩晓冰

(西安科技大学 通信与信息工程学院, 陕西 西安 710054)

摘要: 在机器视觉中需要对采集到的带有畸变的图像进行矫正。为了提高对畸变图像校正的实时性, 针对使用 FPGA 实现图像畸变矫正算法时, 存在的在线计算逆向映射坐标复杂和片上 ROM 容量不够问题, 压缩了逆向映射表, 并在 FPGA 上利用插值方法在线重建了逆向映射表。通过查找重建的逆向映射表来获取逆向映射坐标, 从而降低了 FPGA 的在线计算量和片上 ROM 的容量需求。MATLAB 仿真结果显示, 当压缩参数 n 分别取 4、8、16 时都能够对畸变图像进行较好的矫正, 并且图像信息不会丢失。仿真结果验证了算法的有效性, 该算法可用于基于 FPGA 的图像畸变矫正。

关键词: 机器视觉; FPGA; 图像畸变矫正; MATLAB

中图分类号: TN911.73; TP391.41

文献标志码: A

DOI: 10.5768/JAO202041.0102004

Image distortion correction algorithm based on FPGA

MAO Xinrong, LIU Kaiming, WANG Leyi, HAN Xiaobing

(College of Communication & Information Engineering, Xi'an University of
Science and Technology, Xi'an 710054, China)

Abstract: In machine vision, the collected images with distortion need to be corrected. In order to improve the real-time performance of distorted images correction, aiming at the problem of complex on-line calculation of reverse mapping coordinates and insufficient ROM capacity on chip when using FPGA(Field-Programmable Gate Array) to realize the algorithm of image distortion correction, the compressed reverse mapping table was reconstructed online by interpolation method on FPGA. The reverse mapping coordinates were obtained by seeking the reconstructed reverse mapping table, which reduced the online computation of FPGA and the capacity requirement of ROM on chip. The simulation results of MATLAB show that when the compressed parameters are 4, 8 and 16, respectively, the distortion images can be corrected better and the image information cannot be lost. The simulation results verify the effectiveness of algorithm, which can be used for image distortion correction based on FPGA.

Key words: machine vision; FPGA; image distortion correction; MATLAB

引言

当具有理想形状的透镜与图像传感器保持严格平行时, 摄像头采集的图像不存在任何非线性畸变。但由于制造工艺、安装误差等原因, 现实中不存在理想的透镜, 而且将透镜和图像传感器平面

保持严格平行是非常困难的^[1]。因此, 摄像头真实的成像效果往往会在一定程度的非线性畸变^[2]。含有非线性畸变的图像在一些定量分析, 图像测量领域中会降低图像中物体的几何位置精度^[3], 所以, 在进行相关图像处理之前, 需要完成畸变矫正这一预处理。基于传统的串行处理器实现图像畸

收稿日期: 2019-06-17; 修回日期: 2019-08-19

基金项目: 陕西省重点研发计划 (S2018-YF-YBGY-0041)

作者简介: 毛昕蓉 (1975-), 女, 硕士, 副教授, 硕士生导师, 主要从事宽带网络与光通信技术数字图像处理研究。

E-mail: maoxinrong@xust.edu.cn

变矫正的实时性会随着图像分辨率的提升而变差, 不适合应用在实时性要求高的场景。相比之下, 融合了 ASIC 和基于处理器系统的最大优势的 FPGA, 既能提供硬件定时的速度与稳定性, 其灵活性也与基于处理器系统上运行的软件相当^[4], 非常适合实现图像的实时畸变矫正。因此, 本文针对 FPGA 实现图像畸变矫正算法做了相关的理论研究。

1 畸变矫正算法

基于图像映射的畸变矫正算法由两个基本操作组成: 一是对图像平面上的像素点按照畸变模型进行重新排列; 二是对空间变换后的像素点的灰度值重新赋值^[5]。根据处理数据流向的不同, 重新排列的方式可分为两种^[6]: 一是计算源图像任意像素点经映射后在目标图像中的坐标位置, 即前向映射; 二是计算目标图像任意像素点经映射后在源图像中的坐标位置, 即逆向映射。前向映射和逆向映射的关系如(1)式所示:

$$\begin{aligned} \text{前向映射: } & u = F_u(x, y) \quad v = F_v(x, y) \\ \text{逆向映射: } & x = F_x(u, v) \quad y = F_y(u, v) \end{aligned} \quad (1)$$

式中: (u, v) 是目标图像像素坐标; (x, y) 是源图像像素坐标。相比前向映射, 遍历目标图像的像素点的逆向映射不会出现空点像素问题, 因此像素点的重新排列方式常选用逆向映射。逆向映射坐标的获取可利用相机的内参数矩阵 \mathbf{H} 、畸变系数 $k_i (i=1, 2, 3)$ 和 $p_i (i=1, 2)$ 来计算。假设 (x_p, y_p) 是目标图像上的像素点, (x_s, y_s) 为像素点 (x_p, y_p) 对应的逆向映射坐标, 则逆向映射坐标 (x_s, y_s) 计算如(2)式和(3)式所示^[7]:

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \mathbf{H}^{-1} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 \\ 0 & \frac{1}{dy} \end{bmatrix} \times \begin{bmatrix} x_d(1+k_1r^2+k_2r^4+k_3r^6) + [2p_1y_d + p_2(r^2+2x_d^2)] \\ y_d(1+k_1r^2+k_2r^4+k_3r^6) + [p_1(r^2+2y_d^2) + 2p_2x_d] \\ 1 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} \quad (3)$$

由于经逆向映射得到的坐标一般是非整数, 即计算出的像素坐标不落在原图像的像素点上而是在其附近, 故需要进行插值处理。考虑到不同插值算法的插值效果和硬件实现成本^[8], 本文利用双

线性内插值对空间变换后像素点的灰度值重新赋值, 如图 1 所示。

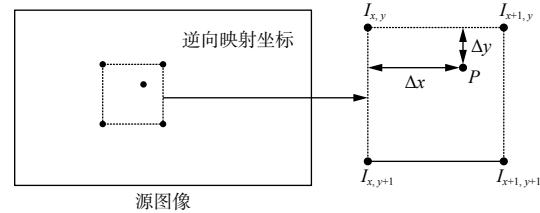


图 1 双线性插值

Fig. 1 Bilinear interpolation

双线性内插值计算过程如(4)式所示, 在两个方向分别进行一次线性插值后, 可得到目标像素的灰度值 $I^{[9]}$ 。

$$\begin{aligned} I_{\text{up}} &= (1 - \Delta x) \cdot I_{x,y} + \Delta x \cdot I_{x+1,y} \\ I_{\text{down}} &= (1 - \Delta x) \cdot I_{x,y+1} + \Delta x \cdot I_{x+1,y+1} \\ I &= (1 - \Delta y) \cdot I_{\text{up}} + \Delta y \cdot I_{\text{down}} \end{aligned} \quad (4)$$

2 基于 FPGA 的图像畸变矫正算法优化

2.1 基于逆向映射表的图像畸变矫正

从(2)式和(3)式中可以看出, 逆向映射坐标 (x_s, y_s) 的计算需要经过非常复杂的运算, 主要包括多阶矩阵求逆, 矩阵与坐标相乘, 各种参数(浮点数)与坐标相乘等运算。若是直接利用 FPGA 在线计算逆向映射坐标, 将会耗费大量的乘法器等资源^[10], 而 FPGA 芯片上如乘法器之类的资源是有限的, FPGA 芯片越低端, 数量越少, 因此乘法器等资源需要节约使用。而且 FPGA 本身也并不适合实现太复杂的运算, 因此利用 FPGA 进行在线计算逆向映射坐标是不可取的。

由于相机的内参数矩阵、畸变系数是固定不变的^[11], 即目标图像到源图像的坐标映射是固定不变的, 因此可先将目标图像的像素点遍历一遍, 计算出各像素点的逆向映射坐标 (x_s, y_s) , 形成一个逆向映射表, 然后将逆向映射表存入 FPGA 的片上 ROM 中。在进行逆向映射时, 只需通过查找片上 ROM 中的逆向映射表就可以获得较高精度的逆向映射坐标, 这比计算一个参数模型快得多, 避免了在 FPGA 上耗费大量的逻辑资源去实现复杂的在线计算。该过程如图 2 所示。

得到逆向映射坐标后, FPGA 还需实现目标像素点的重新赋值。此时可凭逆向映射坐标从帧缓存中获得对应的插值窗口的 4 个角点的灰度值进行双线性内插值计算, 并将计算得到的灰度值赋给目标图像的像素点即可。由(4)式可知, 双线性

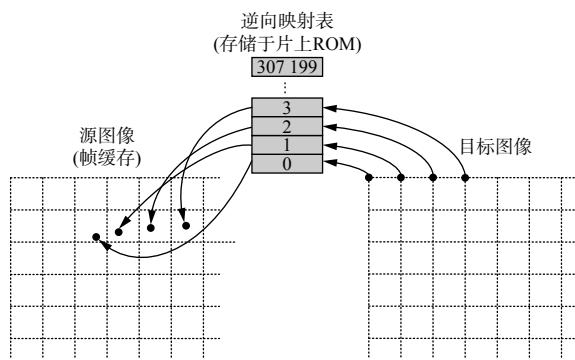


图 2 基于逆向映射表的逆向映射

Fig. 2 Reverse mapping based on reverse mapping table

内插值计算主要是小数与整数的乘法运算和加法运算^[12], 因此 FPGA 可以较小的逻辑资源直接实现双线性内插值计算。

综上所述, 利用 FPGA 实现矫正图像畸变算法的过程如图 3 所示。

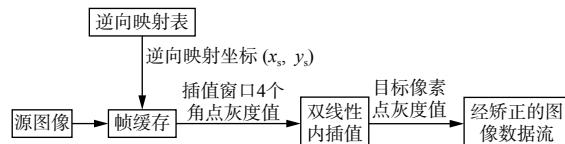


图 3 纠正图像畸变算法的 FPGA 实现

Fig. 3 Implementation of image distortion correction algorithms based on FPGA

图 3 中, 图像畸变矫正算法的 FPGA 实现过程为:

步骤 1: 从逆向映射表(存储于片上 ROM)中查找出与目标图像(经矫正的图像)当前像素点对应的逆向映射坐标(x_s, y_s);

步骤 2: 凭逆向映射坐标(x_s, y_s)从帧缓存中获取对应的插值窗口的 4 个角点的灰度值并进行双线性内插值计算, 将计算得到的灰度值赋给当前像素点;

步骤 3: 跳到目标图像的下一个像素点, 重复步骤 1 和步骤 2, 直到遍历完一帧图像中的所有像素点。

2.2 逆向映射表的存储优化

常见的 FPGA 芯片的片上内存容量普遍较小, 如型号为 Cyclone IV EP4CE115F29 的 FPGA 芯片有 432 个 M9K 内存模块, 总大小约为 3.79 Mb。而需要进行畸变矫正图像尺寸越大, 则对片上内存的容量需求越大。以分辨率为 640×480 像素的图像为例, 其逆向映射坐标(x_s, y_s)中的 x_s (表示列)可由 10 位整数位, 7 位小数位的二进制表示; y_s (表示行)由 9 位整数位, 7 位小数位的二进制表示, 此时逆向映射表的大小如(5)式所示:

$$640 \times 480 \times 33 \text{ bit} = 10137600 \text{ bit} \approx 9.67 \text{ Mb} \quad (5)$$

由(5)式可知, 片上 ROM 无法存储未经压缩的逆向映射表, 针对这个问题, Akin 提出了将逆向映射表存进 DRAM 中, 而郑永瑞则将其固化在 ROM 芯片中, 开始矫正时缓存到 DDR2 芯片中^[13]。但这些方法会增加设计的复杂度并降低处理速度, 为了发挥片上 ROM 的硬件加速功能, 需将逆向映射表压缩至合适的大小。

由于摄像头采集的图像因非线性畸变造成的变形存在着连续性, 即目标图像中毗邻像素点的逆向映射坐标是平滑的且变化缓慢, 因此可仿效图像缩放的原理^[14], 对逆向映射表进行压缩。从计算的角度来看, 最简单的压缩方式就是对逆向映射表进行降采样^[15], 从而降低逆向映射表的分辨率, 使其能够存进片上 ROM, 然后在 FPGA 上通过插值来重建期望的逆向映射表。此时, 图 3 中逆向映射坐标(x_s, y_s)的获取方式如图 4 所示。

图 4 逆向映射坐标(x_s, y_s)的获取Fig. 4 Acquisition of inverse mapping coordinates (x_s, y_s)

1) 基于降采样压缩逆向映射表

以分辨率为 640×480 像素的图像为例, 未经压缩的逆向映射表分辨率为 640×480 像素。图 5 所示为逆向映射表的一部分($n+1$ 行 641 列, 第 641 列为重建需要增加的一列)。

图 5 中, $k=640/n$, 增加的第 641 列是为了保证第 k 个窗口在重建时有一个完整的插值窗口, 第 641 列复制了第 640 列的数值(同理增加第 481 行)。对逆向映射表进行降采样时, 可在水平和垂直方向上每隔 $n-1$ 个存储单元采样一个存储单元存入新的逆向映射表, 此时经过压缩的逆向映射表分辨率 r_{comp} 如(6)式所示, 经压缩的逆向映射表分辨率 r_{comp} 与未压缩的逆向映射表分辨率 r_{ori} 的比例因子 S 如(7)式所示。

$$r_{\text{comp}} = \frac{640+n}{n} \times \frac{480+n}{n} \quad (6)$$

$$S = \frac{r_{\text{ori}}}{r_{\text{comp}}} = \frac{640 \times 480}{\frac{640+n}{n} \times \frac{480+n}{n}} = \frac{640 \times 480}{\frac{307200}{n^2} + \frac{1120}{n} + 1} \quad (7)$$

从(7)式可以看出, 当 n 越大, 比例因子 S 越

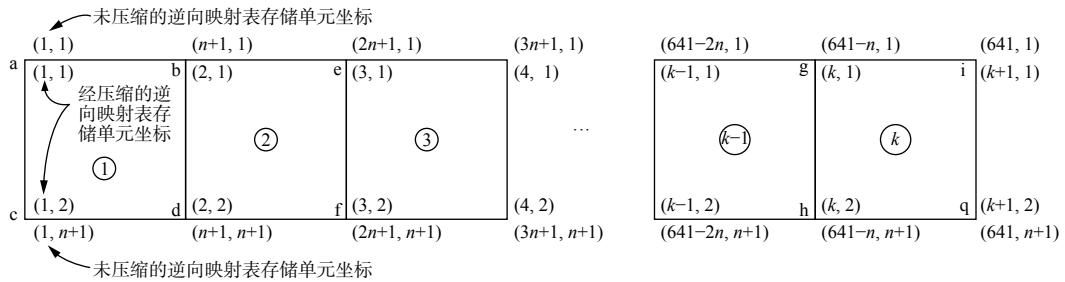


图 5 逆向映射表的压缩

Fig. 5 Compression of reverse mapping table

大, 则压缩程度越大, 逆向映射表对存储空间的需求越低。

2) 基于插值法重建逆向映射表

不同的插值算法重建降采样后的逆向映射表, 所需的采样点数量, 消耗的逻辑资源也不同。3 次内插法所需的采样点数量较少, 但算法复杂度高, 在计算给定位置的映射时所需的逻辑资源较多, 计算速度慢; 双线性内插法所需的采样点数量虽然要多于 3 次内插法, 但由于双线性内插法较简单, 在计算给定位置的映射时所需的逻辑资源较少, 计算速度快。因此, 本文采用双线性内插法来重建期望的逆向映射表, 其原理如图 6 所示。

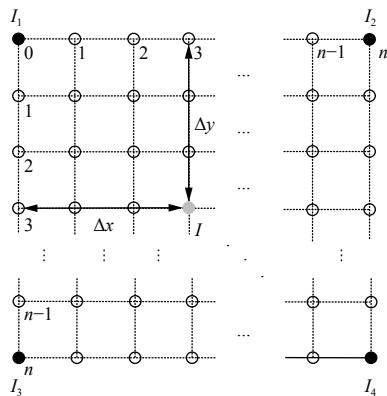


图 6 双线性内插法实现逆向映射表的重建

Fig. 6 Reconstruction of reverse mapping table by bilinear interpolation

由于逆向映射表中的一个存储单元同时存储了 x_s 和 y_s , 故 x_s 和 y_s 的值需分别计算。以计算 x_s 为例, 图 6 中黑色的点 I_1 、 I_2 、 I_3 和 I_4 为降采样后的逆向映射表中的 4 个 x_s , 灰色点 I 和空心点都为重建逆向映射表时数值未知的 x_s 。设点 I_1 、 I_2 、 I_3 和 I_4 之间的水平和垂直距离都为 n , Δx 为点 I 的水平方向权值, Δy 为点 I 的垂直方向权值, 利用双线性内插法去拟合 4 个点 I_1 、 I_2 、 I_3 和 I_4 之间的 $(n+1)^2-4$ 个 x_s 的数值, 以图中的点 I 为例, 其计算

过程如(8)式所示:

$$\begin{aligned} I_{\text{up}} &= ((n-\Delta x) \cdot I_1 + \Delta x \cdot I_2) / n \\ I_{\text{down}} &= ((n-\Delta x) \cdot I_3 + \Delta x \cdot I_4) / n \\ I &= ((n-\Delta y) \cdot I_{\text{up}} + \Delta y \cdot I_{\text{down}}) / n \end{aligned} \quad (8)$$

从(8)式可以看出, 当在 FPGA 上利用双线性内插法在线重建逆向映射表时, 需要实现除数为 n 的除法运算。在 FPGA 上实现除数不等于 2^i ($i=1, 2, 3, \dots$) 的除法会耗费较多的逻辑资源, 但如果除数为 2^i ($i=1, 2, 3, \dots$) 时, 硬件实现上只需要做右移 i 位的操作便可实现除法, 因此当 n 等于 2^i ($i=1, 2, 3, \dots$) 时, 硬件实现双线性内插法计算的成本最小。而经降采样后逆向映射表的大小为原来的 $1/S$ 。

3) 优化后算法的分析

基于坐标映射的图像畸变矫正算法是目前最为成熟和有效的畸变矫正方法。但基于不同插值方法的映射算法在图像质量, 资源消耗等方面都存在一定差异^[16], 这些差异会影响最终在 FPGA 上实现的效果。常用的插值方法有最邻近元法和 3 次内插值方法。最邻近元法虽然速度快, 资源使用量小, 但是矫正后的图像质量不佳, 易产生锯齿失真。3 次内插值法能够得到较好的图像质量, 但是算法代价过大, 会消耗大量的加法器和乘法器资源, 十分不适于在 FPGA 上进行实现。本文所采用的双线性内插方法结合了两者的优点, 在保证图像质量的同时, 有效降低了算法开销和逻辑资源占用, 使之能更好地在 FPGA 平台上进行实现。表 1 给出了基于几种不同插值算法的畸变矫正算法的比较。

表 1 3 种畸变矫正算法的比较

Table 1 Comparison of three distortion correction algorithms

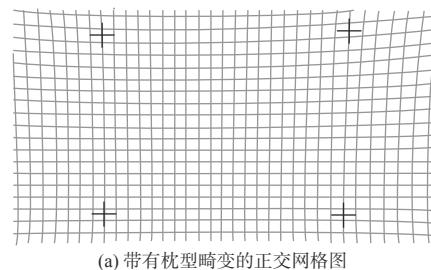
插值方法	基于最邻近元法	基于双线性内插法	基于3次内插法
图像质量	有锯齿失真清晰度不高	清晰度较高	清晰度最好
算法开销	最低	较低	最高
逻辑资源需求	低	较低	高

3 实验结果

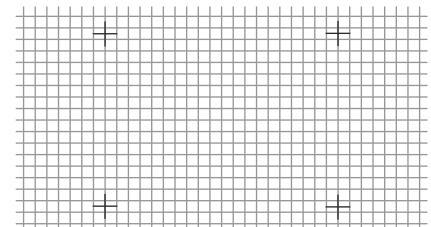
为了验证本文提出方法的有效性, 采用了一个含有枕型畸变的正交网格图作为待矫正图像, 采用本文的方法对其进行矫正, 其结果如图 7 所示。从结果可以看出, 相较于图 7(a) 中含有枕型畸变的正交网格线, 经过本文方法矫正后的图像效果改善明显, 说明了本文提出的方法能够很好地对图像畸变进行矫正。

为了进一步说明通过压缩逆向映射表来提高算法性能后对矫正结果的影响, 本文在 MATLAB 上实现了逆向映射表(分辨率为 640×480 像素)的压缩, n 的值取 4、8、16 和 32, 然后利用经双线性内插法在线重建的逆向映射表对一幅含桶型畸变的图像进行矫正, 矫正效果如图 8 所示。

图 8(a) 的图像是一幅由摄像头采集的含有桶型畸变的原始图像。图 8(b) 中图像为使用不经压



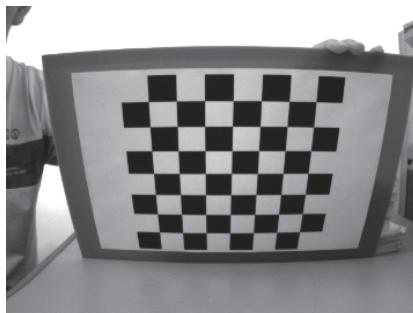
(a) 带有枕型畸变的正交网格图



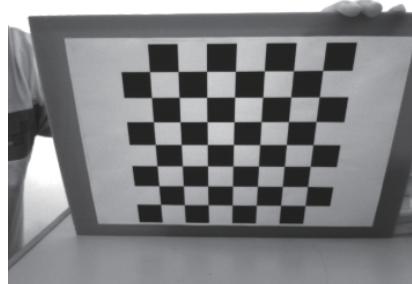
(b) 使用本文方法对图像矫正后的结果

图 7 带有枕型畸变的正交网格图矫正结果

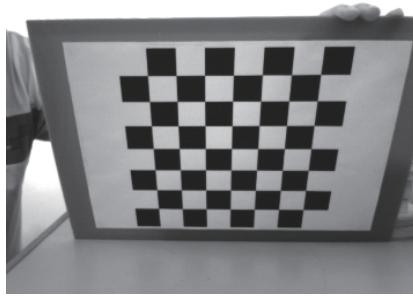
Fig. 7 Correction result of orthogonal grids with pillow-type of distortion



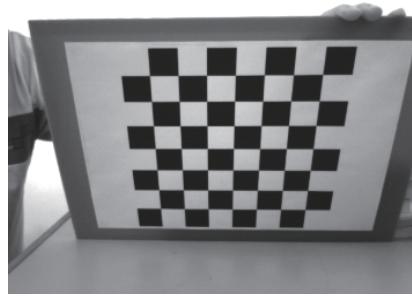
(a) 含畸变原图



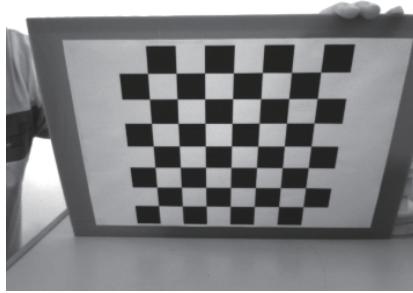
(b) 标准矫正效果图



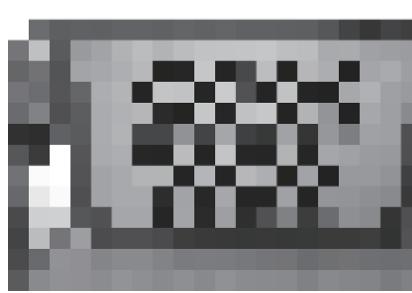
(c) $n=4$



(d) $n=8$



(e) $n=16$



(f) $n=32$

图 8 矫正效果对比

Fig. 8 Comparison of correction effect

缩的逆向映射表进行矫正后所得的图像。对比图 8(a) 和 8(b) 可以明显观察到经过矫正后图形中的桶型畸变已经消失, 标定板的边缘已经由外凸变为水平, 说明该矫正算法能够很好地对畸变图像进行矫正。

另外, 对比图 8(a)、8(c)、8(d)、8(e) 几幅图片可以发现, 当通过对参数 n 分别取 $n=4$ 、 $n=8$ 、 $n=16$ 时对逆向映射表进行压缩之后, 使用本文中的算法对图像畸变进行矫正, 其结果也较为理想。在图 8(c)、8(d)、8(e) 中桶形畸变都被明显矫正, 且矫正后的图像效果清晰, 证明了可以通过压缩逆向映射表

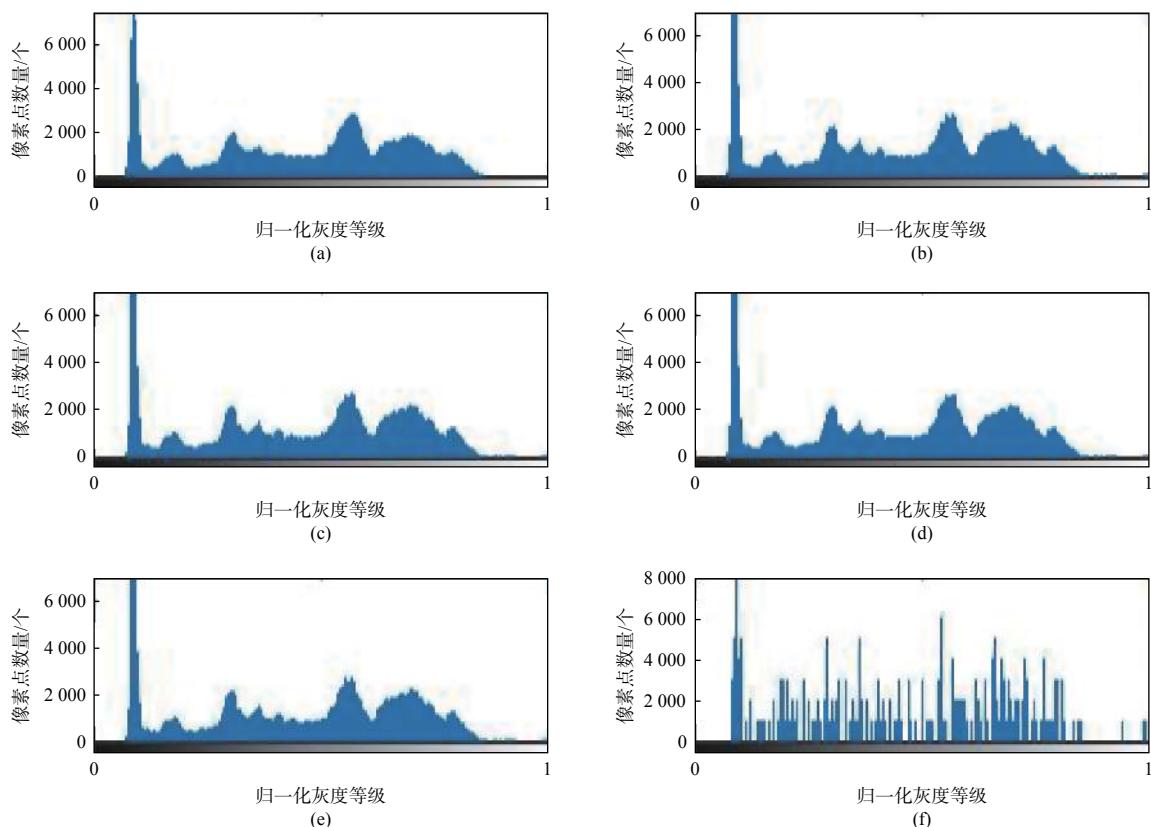


图 9 灰度直方图

Fig. 9 Gray histogram

为了分析压缩图像映射表对于矫正后图像的影响情况, 以图 8(b) 中的图像作为标准图像, 分别计算 8(c)、8(d)、8(e)、8(f) 中图像与 8(b) 中图像的 PSNR 与 SSIM。以此来衡量图像质量的变化情况, 结果如表 2 所示。

表 2 中, 当 PSNR 高于 40 dB 时, 说明图像非常接近标准矫正图像, 质量非常好; 低于 20 dB 时, 说明图像质量很差, 已不可接受。而 SSIM 值越高, 则表明图像失真越小。

由上述的仿真结果可知, 适当降采样后的逆向

的方式在得到清晰图像的同时节省 FPGA 的硬件资源, 并减小算法复杂度。

值得注意的是, 在图 8(f) 中, 当 $n=32$ 时图像出现了严重的失真。为了对该结果进行分析, 绘制了图 8 中图像的灰度直方图, 如图 9 所示。通过观察图 9 中的直方图可以发现, 由于畸变矫正只是矫正了图像的部分信息, 纠正前后图像的基本特征是一致的。因此, 畸变图像与经过有效矫正后的图像拥有近似的灰度直方图分布。但是当 $n=32$ 时, 纠正后图像的像素分布已经严重偏离原始图像, 出现了马赛克现象, 细节严重丢失。

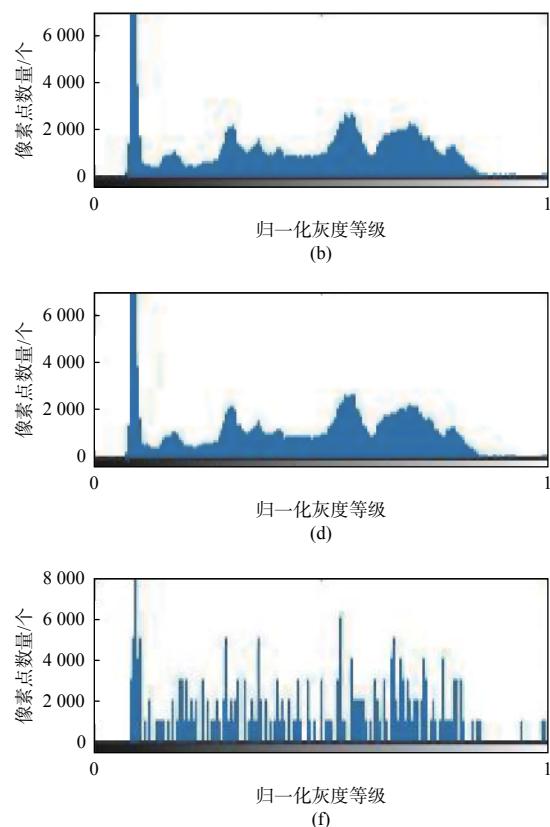


图 9 灰度直方图

Fig. 9 Gray histogram

映射表可以很好地实现图像畸变矫正, 能够在保证图像清晰度的情况下将逆向映射表压缩为原来

表 2 利用经压缩的逆向映射表矫正畸变图像的 PSNR 和 SSIM

Table 2 Correction of PSNR and SSIM of distorted image using compressed reverse mapping table

n	4	8	16	32
PSNR/dB	61.866 5	58.241 2	54.230 5	11.539 3
SSIM	0.999 8	0.999 5	0.999 0	0.599 9

的 1/16。但通过对逆向映射表进行降采样再利用插值重建并不能无限制地减小其分辨率, 其分辨率的大小取决于逆向映射的平滑度和重建所需的精度。

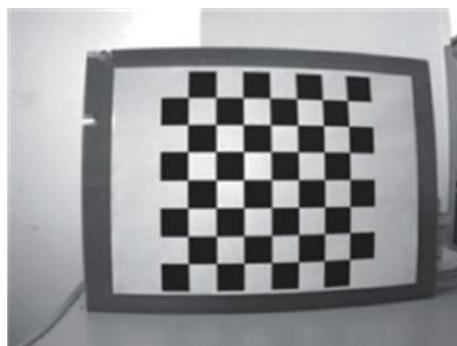
通过搭建 FPGA 双目视觉采集系统, 验证本文算法的硬件可实现性。选用 Intel FPGA Cyclone IV 系列 EP4CE115F29C7 芯片作为图像采集及预处理系统的核心, 如图 10 所示。使用两颗 CMOS 图像传感器进行采集, 并在板上通过本文算法进行处理。输出结果显示, 采用本文算法后, 能够使得图像畸变得到矫正, 提高

图像质量以满足后续图像处理的需求, 如图 11 所示。



图 10 FPGA 双目视觉采集系统

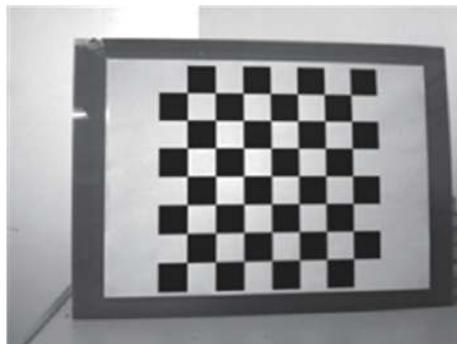
Fig. 10 Binocular vision acquisition system based on FPGA



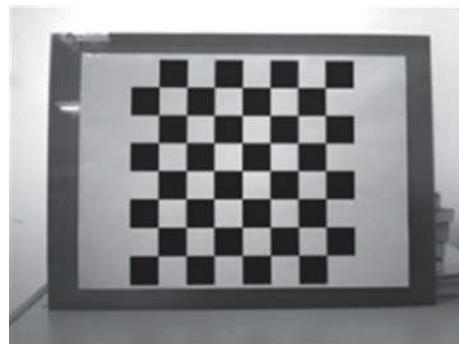
(a) 原始左目图像



(b) 原始右目图像



(c) 纠正后左目图像



(d) 纠正后右目图像

图 11 双目视觉采集系统效果图

Fig. 11 Effect diagram of binocular visual acquisition system

4 结论

为了提高矫正含畸变图像的实时性, 本文研究了如何在 FPGA 上实现畸变矫正算法。针对 FPGA 实现图像畸变矫正算法时, 存在线计算逆向映射坐标复杂和片上 ROM 容量不够的问题, 对图像畸变矫正算法进行了优化, 即通过查找逆向映射表和对逆向映射表进行降采样再利用插值重建, 降低了在线计算量和片上 ROM 的容量需求。实验结果表明, 在合适的分辨率下, 优化后的畸变矫正算法具有良好的矫正效果, 并具有广泛的实际应用价值。

参考文献:

- [1] GUAN Wei, WANG Zhangli, WANG Zhongqiang, et al. Precise axis centering calibration technology for optical lens[J]. Journal of Applied Optics, 2019, 40(2): 223-228. 管伟, 王章利, 王中强, 等. 光学镜头轴线精确定心校准技术[J]. 应用光学, 2019, 40(2): 223-228.
- [2] PAN Bing, YU Liping, WU Dafang, et al. Systematic errors in two-dimensional digital image correlation due to lens distortion[J]. Optics and Lasers in Engineering, 2013, 51: 140-147.
- [3] DUAN Mali. Research and implementation of distortion

- correction algorithms for wide angle images[D]. Handan: Hebei University of Engineering, 2013.
- 段马丽. 广角图像畸变校正算法的研究与实现[D]. 邢: 河北工程大学, 2013.
- [4] WU Song, ZHANG Lihua. Design and implementation of image processing system based on FPGA [J]. Radio Communication Technology, 2018, 44(4): 407-410.
- 吴松, 张利华. 基于FPGA的图像处理系统设计与实现[J]. 无线电通信技术, 2018, 44(4): 407-410.
- [5] LIN Minghan, HU Xianlong, HUANG Xing, et al. Image distortion correction of CMOS with large field[J]. Science Technology and Engineering, 2009, 9(18): 5508-5510.
- 林明汉, 胡贤龙, 黄星, 等. 大视场CMOS相机图像的畸变校正[J]. 科学技术与工程, 2009, 9(18): 5508-5510.
- [6] LU Yufeng, LUO Xiaohua, YU Miao, et al. Line buffer reduction for LUT-based real time image rectification[J]. Journal of Fudan University : Natural Science, 2016, 55(4): 418-424.
- 卢宇峰, 罗小华, 俞淼, 等. 基于LUT实时图像矫正的行缓存优化[J]. 复旦学报: 自然科学版, 2016, 55(4): 418-424.
- [7] HAO Renjie, WANG Zhongyu, LI Yaru. Error analysis method for monocular vision pose measurement system[J]. Journal of Applied Optics, 2019, 40(1): 79-85.
- 郝仁杰, 王中宇, 李亚茹. 一种单目视觉位姿测量系统的误差分析方法[J]. 应用光学, 2019, 40(1): 79-85.
- [8] CHEN Gaolin. Comparison of common interpolation methods in image scaling algorithms[J]. Journal of Fujian Computer, 2017, 33(9): 98-99.
- 陈高琳. 图像缩放算法中常见插值方法比较[J]. 福建电脑, 2017, 33(9): 98-99.
- [9] XU Pengfei. Application and realization of bilinear interpolation and cubic convolution image scaling[J]. Network Security Technology & Application, 2017(12): 51-67.
- 徐鹏飞. 双线性插值和三次卷积在图像缩放中的应用及实现[J]. 网络安全技术与应用, 2017(12): 51-67.
- [10] XU Fang, XI Yi, CHEN Hong, et al. Design and implementation of matrix hardware acceleration based on FPGA/ Nios- II [J]. Journal of Electronic Measurement and Instrumentation, 2011, 25(4): 377-383.
- 许芳, 席毅, 陈虹, 等. 基于FPGA/Nios- II 的矩阵运算硬件加速器设计[J]. 电子测量与仪器学报, 2011, 25(4): 377-383.
- [11] HAO Meilin, GAO Minghui, LI Shuang, et al. Research on a monocular camera calibration algorithm[J]. Digital Communication World, 2018(5): 11-14.
- 赫美琳, 高明慧, 李爽, 等. 一种单目相机标定算法研究[J]. 数字通信世界, 2018(5): 11-14.
- [12] SHAO Zhengfen. FPGA implementation of best-precision fixed-point algorithm[J]. Communication Technology, 2009, 42(7): 279-281.
- 邵正芬. 最佳精度定点运算的FPGA实现[J]. 通信技术, 2009, 42(7): 279-281.
- [13] ZHENG Yongrui, LI Jie, LIU Xiaohong, et al. FPGA based real-time geometric distortion correction for video images[J]. Electronics Optics & Control, 2013, 20(6): 75-78.
- 郑永瑞, 李洁, 刘晓宏, 等. 基于FPGA的视频图像实时几何畸变校正[J]. 电光与控制, 2013, 20(6): 75-78.
- [14] ZHANG Yang. Design and implementation of image scaling algorithm based on bilinear interpolation[J]. Electronic Design Engineering, 2016, 24(3): 169-170.
- 张洋. 基于双线性插值法的图像缩放算法的设计与实现[J]. 电子设计工程, 2016, 24(3): 169-170.
- [15] SONG Shoupeng, SHAO Yonghua, DU Ying. Survey of sampling method[J]. Journal of Data Acquisition and Processing, 2016, 31(3): 452-463.
- 宋寿鹏, 邵勇华, 堵莹. 采样方法研究综述[J]. 数据采集与处理, 2016, 31(3): 452-463.
- [16] DIANA EARSHIA V, SUMATHI M. A comprehensive study of 1D and 2D image interpolation techniques[M]. Singapore: Springer Singapore, 2018: 383-391