



基于残余插值的卷积神经网络去马赛克算法

摘要

在去马赛克问题中,为了精确插值倾斜边缘并提高结果图像的整体质量,提出一种基于残余插值的卷积神经网络去马赛克算法.针对 Bayer 格式的颜色滤波阵列,插值绿色平面时,对于红蓝通道信息不全的问题,采用同通道邻近像素值近似代替,综合考虑 3 个通道的梯度,运用倾斜方向的边缘检测算子,将倾斜边缘分为不同方向的边缘分别插值.在插值完成后,利用深度卷积神经网络,进一步训练插值结果.在标准的 IMAX 数据集上,与目前流行的算法相比,本文算法视觉上更接近原图,具有更高的峰值信噪比和更短的运行时间.

关键词

去马赛克; Bayer CFA 模式; 残余插值; 边缘检测; 卷积神经网络

中图分类号 TP391.41

文献标志码 A

收稿日期 2017-08-28

资助项目 国家自然科学基金(61372184);北京市自然科学基金(4112061)

作者简介

周登文(通信作者),男,教授,研究方向为图像处理和计算机视觉. zdw@ncepu.edu.cn

1 华北电力大学 控制与计算机工程学院,北京,102206

0 引言

单传感器彩色成像在数码相机行业应用广泛.在一个单传感器相机里,传感器表面覆盖一层彩色滤波阵列(CFA, Color Filter Array).每个像素点仅采样红、绿或蓝色分量之一,要恢复全彩色图像,丢失的 2 种颜色分量需要估计,这个估计过程,称为去马赛克.

本文针对应用最广泛的 Bayer CFA 模式^[1](图 1),其他形式 CFA 可参考文献[2-3].在 Bayer CFA 模式中,绿色像素点按梅花形网格采样,红色和蓝色像素点按矩形网格采样,绿色样本是红色或蓝色样本数目的 2 倍.

到目前为止,已提出了许多去马赛克算法^[4].去马赛克过程是一个图像插值的过程,双线性插值和双三次等简单的插值方法,在图像平滑区域也有较好的效果.但是它们相当于各向同性的低通滤波,会产生显著的边缘模糊、拉链效应等缺陷.

Hamilton 等^[5]提出自适应彩色平面插值算法,利用二阶导数检测局部水平和垂直边缘的方向,然后,分别沿着水平和垂直的边缘方向进行插值,改进了双线性等简单插值方法的结果.该算法利用了颜色通道之间的相关性,将红色和蓝色的二阶梯度作为修正因子来计算绿色分量.恢复出绿色分量后,再以绿色分量的二阶梯度作为修正因子来恢复红色和蓝色分量.

由于颜色通道之间具有相关性,如果绿色通道某区域有边缘,那么同区域的红色和蓝色通道上有边缘的概率极大,2 个通道的差图像更平滑,插值效果更好.基于此观察,Peckucuksen 等^[6]提出了基于梯度的无阈值插值算法(GBTf),利用文献[5]的插值公式对绿色平面方向插值之后,计算水平和垂直方向的颜色差,从上下左右 4 个方向对颜色差进行混合,得到最终的色差估计,色差估计与 CFA 中的红色或蓝色分量相加得到插值结果.

沿用色差插值思想,如果在比颜色差更平滑的平面上插值,能更容易恢复图像.Kiku 等^[7]的实验证明,引导滤波方法^[8]生成的估计值与真实像素值的差,即残余,比颜色差更加平滑.他们提出的残余插值(RI)^[7]改进了 GBTf 算法,将 GBTf 中第 1 步的方向插值改为残余插值,然后同样计算并混合颜色差得到最后的结果.残余插值恢复出的图像视觉上更接近原图,并且具有更高的峰值信噪比,是目前去马赛克效果最好的算法之一.

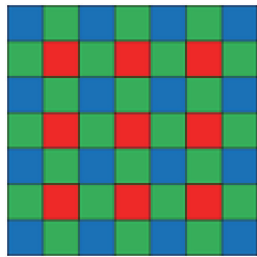


图1 Bayer 颜色滤波阵列^[1]
Fig.1 Bayer color filter array^[1]

本文受残余插值^[7]启发,提出基于残余插值的卷积神经网络算法(RICNN).恢复绿色通道时提出边缘检测算法,能同时检测水平垂直和对角线方向的边缘.实验证明该策略在具有斜边的区域效果显著优于其他算法.恢复出的绿色平面作为红蓝平面的引导图,再次采用残余插值方法恢复红蓝平面.在对残余域线性插值时,本文提出一种类 Laplacian 插值模板,可以保持边缘信息进而对图像产生锐化效果,得到的去马赛克图像边缘更清晰.

得到残余插值结果后,本文用卷积神经网络(CNN,Convolutional Neural Networks)^[9]作为一个修正项进一步改进结果图像.将去马赛克结果作为卷积神经网络的输入,对应的全彩色图像和去马赛克结果之间的残余作为卷积神经网络的标签,经过训练逐步修正卷积神经网络的权重.实验证明,本文算法得到的结果在较小的计算开销内,显著提升了结果图像的质量.

1 基于残余插值的卷积神经网络算法

为了便于介绍本文方法,先简单介绍一下残余插值(图2).以插值红色平面为例说明算法流程.首先线性插值生成绿色平面,以此绿色平面为引导图,应用引导滤波方法^[8]生成红色平面的初步估计值,该初步估计值与原始数据作差得到残余域.在残余域上插值得到的结果再加上初步估计值生成红色平面的插值结果.蓝色和绿色平面插值过程与红色类似,在此不过多赘述.

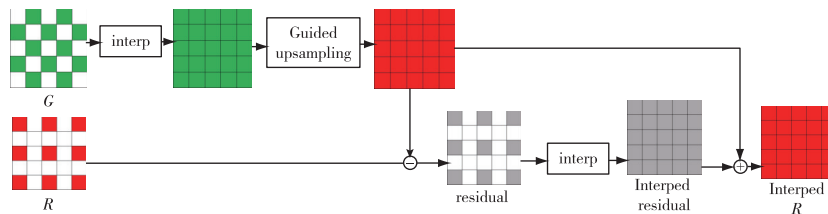


图2 残余插值
Fig.2 Residual interpolation

目前去马赛克的普遍做法是,判断水平或垂直方向的边缘,根据判断结果确定插值方向,但实际中的图像,一般存在各种方向的边.基于此观察,在插值绿色分量时,本文加入倾斜方向的边缘检测算子,采用文献的方法利用局域内颜色差的和判断水平和垂直边缘.如果水平和垂直方向均无边,则判断倾斜方向是否存在边缘,若每个方向均无边,则混合水平和垂直方向的插值结果.

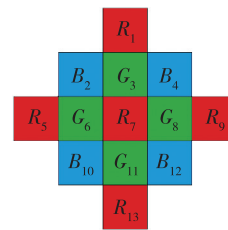


图3 在红色采样点处插值绿色分量
Fig.3 Interpolate G at sampled R

如图3所示,以在 R_7 处插值绿色分量 G_7 为例,该位置处倾斜方向的边分为4种情况,分别为 $G_{3,6}$ 方向、 $G_{3,8}$ 方向、 $G_{6,11}$ 方向、 $G_{8,11}$ 方向.沿4个方向计算梯度,如果4个梯度中的最小值与其平均值之间的比值小于某个阈值,则判定为梯度最小的这个方向有倾斜的边缘,插值沿此方向进行.

时,与双线性相比,考虑在更大的模板上插值,这样能够更充分地利用图像信息,同时为了更好地保持边缘信息,受 Laplacian 滤波器启发,本文提出一种类 Laplacian 滤波器(式(5)).在考虑了更多图像信息的同时,能够产生锐化的效果,使结果图像边缘比双线性更清晰.

在恢复红蓝平面时,以插值过的绿色平面为引导图,同样用残余插值方法.在对残余域线性插值

文献[9]中的卷积神经网络(CNN)用于解决超

分辨率问题,能够将低分辨率图像恢复为分辨率较高的图像.本文受此启发,在去马赛克问题中,将插值结果看作分辨率较低的图像,用卷积神经网络作为一个后处理的步骤,对低分辨率图像进行修正,进而提高插值精度.

1.1 算法介绍

1) 对训练图像去马赛克

与大多数算法类似,首先插值绿色平面.插值绿色平面的流程如图4所示.第1步,分别在水平和垂直方向用 MLRI 方法对 R, G, B 进行插值得到 $\tilde{R}^H, \tilde{G}^H, \tilde{B}^H, \tilde{R}^V, \tilde{G}^V, \tilde{B}^V$.第2步,在水平和垂直方向分别计算色差($\tilde{\Delta}_{g,r}^H = \tilde{G}^H - \tilde{R}^H$);第3步,做决策.在 5×5 窗口内对色差的绝对值求和得到 σ_h, σ_v .令 $\sigma_1 = \max(\sigma_h/\sigma_v, \sigma_v/\sigma_h)$,用来检测水平和垂直方向的边.如果水平和垂直方向均检测不到边,则在斜方向上检测边,如图3所示, R_2 处的绿色分量在倾斜方向的边可能有4种情况,分别为 $G_{3,6}$ 方向、 $G_{3,8}$ 方向、 $G_{6,11}$ 方向和 $G_{8,11}$ 方向,式(2)和式(3)分别定义这4个方向的梯度和估计值,在 5×5 窗口内对梯度求和,得到 $\sigma_{3,6}, \sigma_{3,8}, \sigma_{6,11}, \sigma_{8,11}$,令 $m = \min(\sigma_{3,6}, \sigma_{6,8}, \sigma_{6,11}, \sigma_{8,11})$, $n = \text{mean}(\sigma_{3,6}, \sigma_{3,8}, \sigma_{6,11}, \sigma_{8,11})$, $\sigma_2 = m/n$,以此作为倾斜方向的边缘检测算子.给定阈值 $\tau_1 = 1.9, \tau_2 = 0.4$,采用式(1)中策略插值绿色像素,其中方向权重通过对原图求梯度得到 d^H, d^V ,取梯度的倒数来获得.

$$\begin{cases} \text{if } (\sigma_1 > \tau_1) \& (\sigma_1 = \sigma_v/\sigma_h), G_{i,j} = \tilde{G}_{i,j}^H, \\ \text{else if } (\sigma_1 > \tau_1) \& (\sigma_1 = \sigma_h/\sigma_v), G_{i,j} = \tilde{G}_{i,j}^V, \\ \text{else if } (\sigma_2 > \tau_2) \& (m = \sigma_{3,6}), G_{i,j} = \tilde{G}_{i,j}^{3,6}, \\ \text{else if } (\sigma_2 > \tau_2) \& (m = \sigma_{3,8}), G_{i,j} = \tilde{G}_{i,j}^{3,8}, \end{cases}$$

$$\begin{cases} \text{else if } (\sigma_2 > \tau_2) \& (m = \sigma_{6,11}), G_{i,j} = \tilde{G}_{i,j}^{6,11}, \\ \text{else if } (\sigma_2 > \tau_2) \& (m = \sigma_{8,11}), G_{i,j} = \tilde{G}_{i,j}^{8,11}, \\ \text{else } G_{i,j} = \frac{w^H \tilde{G}_{i,j}^H + w^V \tilde{G}_{i,j}^V}{w^H + w^V}, \end{cases} \quad (1)$$

$$\begin{cases} d_{3,6} = |G_3 - G_8| + |G_6 - G_{11}| + |2R_7 - R_1 - R_5| + \\ \quad |2B_2 - B_4 - B_6|, \\ d_{3,8} = |G_3 - G_8| + |G_8 - G_{11}| + |2R_7 - R_1 - R_9| + \\ \quad |2B_4 - B_2 - B_{12}|, \\ d_{6,11} = |G_3 - G_6| + |G_8 - G_{11}| + |2R_7 - R_5 - R_{13}| + \\ \quad |2B_{10} - B_2 - B_{12}|, \\ d_{8,11} = |G_3 - G_8| + |G_6 - G_{11}| + |2R_7 - R_9 - R_{13}| + \\ \quad |2B_{12} - B_4 - B_{10}|, \end{cases} \quad (2)$$

$$\begin{cases} \tilde{G}^{3,6} = (G_3 + G_6)/2 + (2R_7 - R_1 - R_5)/4, \\ \tilde{G}^{3,8} = (G_3 + G_8)/2 + (2R_7 - R_1 - R_9)/4, \\ \tilde{G}^{6,11} = (G_6 + G_{11})/2 + (2R_7 - R_5 - R_{13})/4, \\ \tilde{G}^{8,11} = (G_8 + G_{11})/2 + (2R_7 - R_9 - R_{13})/4, \\ d^H = |G_{i,j} - 1| - |G_{i,j+1}| + |2R_{i,j} - R_{i,j-2} - R_{i,j+2}|, \\ d^V = |G_{i-1,j}| - |G_{i+1,j}| + |2R_{i,j} - R_{i-2,j} - R_{i+2,j}|. \end{cases} \quad (3)$$

插值完的绿色平面作为红色和蓝色平面的引导图帮助恢复红蓝平面,这里同样采用残余插值方法.以插值红色平面为例,蓝色平面与其进行相同处理,先引导滤波生成红色平面的初步估计值,然后在红色采样点处计算估计值和真实像素值之间的残余,对残余线性插值,残余插值结果与估计值相加得到最后的红色平面.对蓝色和绿色采样点处的残余域线性插值采用式(5)中的插值模板.

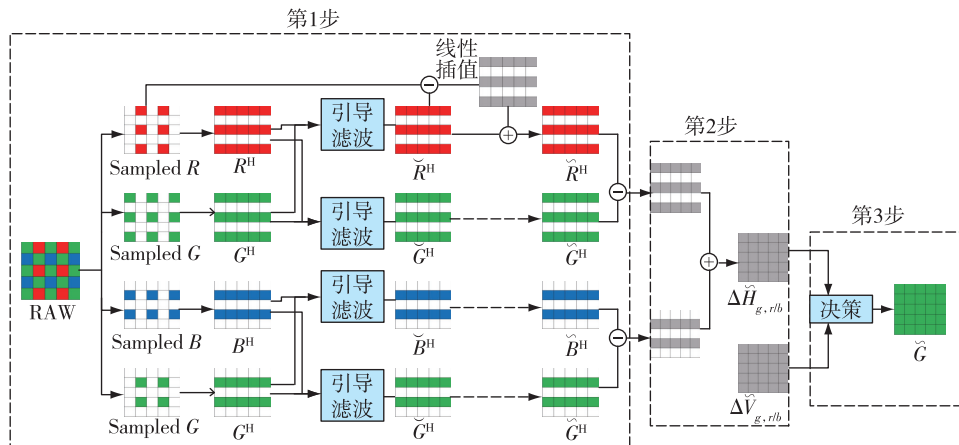


图4 插值绿色分量流程

Fig. 4 Outline of the proposed G pixel value interpolation

$$\begin{cases}
 P_b = \begin{bmatrix} 1 & 0 & -9 & 0 & -9 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 81 & 0 & 81 & 0 & -9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 81 & 0 & 81 & 0 & -9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -9 & 0 & -9 & 0 & 1 \end{bmatrix} \times \frac{1}{256}, \\
 P_g^H = [-16 \ 0 \ 144 \ 0 \ 144 \ 0 \ -16] \times \frac{1}{256}, \\
 P_g^V = [-16 \ 0 \ 144 \ 0 \ 144 \ 0 \ -16]^T \times \frac{1}{256}, \\
 \begin{cases}
 \Delta \tilde{R}_{b_i,j} = \Delta R(i-3:i+3, j-3:j+3) \otimes P_b, \\
 \Delta \tilde{R}_{g_i,j}^H = \Delta R(i, j-3:j+3) \otimes P_g^H, \\
 \Delta \tilde{R}_{g_i,j}^V = \Delta R(i-3:i+3, j) \otimes P_g^V.
 \end{cases}
 \end{cases} \quad (5)$$

2) 用卷积神经网络修正去马赛克结果

经过去马赛克过程,得到相对分辨率较低的图像 Y ,对 n 张训练图像去马赛克可以得到低分辨率数据集 $Y_i, i \in 1, \dots, n$,其对应的高分辨率数据集为 $X_i, i \in 1, \dots, n$,以 Y_i 为输入, $X_i - Y_i$ 为标签训练卷积神经网络.网络第 1 层对 Y 有重叠地提取小片并表示成高维向量;第 2 层实现由低维向量到高维向量的非线性映射,映射出的高维向量代表高分辨率小块,这些向量包含另一组特征映射;第 3 层由高维向量重建图像,代表高分辨率小块的高维向量重构为高分辨率小块.

3 层网络结构形式化表示如下:

$$F_1(Y) = \max(0, W_1 * Y + B_1), \quad (7)$$

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2), \quad (8)$$

$$F(Y) = W_3 * F_2(Y) + B_3, \quad (9)$$

其中 W_i, B_i 分别表示卷积核和偏置项, $*$ 表示卷积操作. W_1 包含 n_1 个大小为 $c \times f_1 \times f_1$ 大小的滤波器, c 是输入图像的通道数, f_1 表示每个滤波器的空间大小.在图像上进行 n_1 次卷积操作,所有的滤波器尺寸为 $c \times f_1 \times f_1$. B_1 是 n_1 维向量,向量中每个元素对应一个滤波器.网络第 1 层实现为每个小片提取到 n_1 维特征. W_2 包含 n_2 个大小为 $n_1 \times f_2 \times f_2$ 的滤波器, B_2 是 n_2 维的偏置项,输出的 n_2 维向量代表高分辨率小块. W_3 包含 c 个大小为 $n_2 \times f_3 \times f_3$ 的滤波器, B_3 是 c 维向量.

1.2 网络训练

学习映射函数 F 需要估计网络参数 $\theta = \{W_1, W_2, W_3, B_1, B_2, B_3\}$,这通过最小化重建图像 $F(Y; \theta)$ 和对应的真实的高分辨率和低分辨率图像的差,即

$X - Y$ 之间的损失函数来得到.给定一组高分辨率图像 $\{X_i\}$ 和对应的低分辨率图像 $\{Y_i\}$,用最小化均方误差(MSE)作为损失函数:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \|F(Y_i; \theta) - (X_i - Y_i)\|^2, \quad (10)$$

n 是训练样本数.用 MSE 做损失函数可以得到更高的峰值信噪比.峰值信噪比被广泛用于图像重建质量的评估,是反映图像恢复质量好坏的客观依据.损失函数用随机梯度下降法最小化,权重按以下方式更新:

$$\Delta_{i+1} = 0.9 \cdot \Delta_i + \eta \cdot \frac{\partial L}{\partial W_i^l}, \quad W_{i+1}^l = W_i^l + \Delta_{i+1}, \quad (11)$$

其中 $l \in \{1, 2, 3\}$, i 是层数索引, η 是学习率, $\frac{\partial L}{\partial W_i^l}$ 是导数.每一层滤波器的权重用高斯分布随机地初始化,该高斯分布以 0 为均值, 0.001 为标准差.前 2 层学习率为 10^{-4} ,最后 1 层学习率为 10^{-5} .

训练过程中,从训练图像中随机有重叠地采样真实图像 $\{X_i\}$ 为 $f_{\text{sub}} \times f_{\text{sub}} \times c - p_{\text{ixel}}$ 小片,把这些小片当做一幅幅小的图像来进行处理.为了得到网络中输入的低分辨率图像,首先对真实图像采样为 Bayer CFA 模式的图片,然后对其进行去马赛克处理.

为了消除边缘影响,训练过程中所有的卷积层没有填充,网络产生一个更小的输出 $((f_{\text{sub}} - f_1 - f_2 - f_3 + 3)^2 \times c)$.

2 实验结果

本文实验中 Bayer 格式的图像通过有间隔地从真实图像中采样得到.采用文献[10]中 91 幅图像作为训练集,随机裁剪为 33×33 的小块,并用本文的去马赛克方法产生低分辨率样本集,在 Caffe 上训练模型.选择 IMAX 数据集^[11]作为测试集测试模型.设置权重衰减项为 0,动力为 0.9,采用随机梯度下降的优化策略.卷积网络的各层参数为 $n_1 = 64, n_2 = 32, f_1 = 9, f_2 = 5, f_3 = 5$.本文实验环境为 GeForce GTX TITAN GPU, 32 GB 内存, ubuntu 操作系统和 Matlab 16.04(R2016a)平台.

图 5 是本实验用来测试去马赛克性能的标准 IMAX18^[11]数据集. IMAX 数据集包含 18 张尺寸 500×500 的图像.图 6 展示了对 IMAX 数据集中的图像利用 RICNN 算法去马赛克和其他去马赛克方法在视觉上的比较,其中图 6a 和 6i 分别是原图和本文提出的算法,可以看出本文算法在倾斜边缘处明显减少了拉链效应,与现有算法相比在视觉上更接近原图.算法在尺寸为 500×500 图像上去马赛克运行时间为 2.88 s.



图5 IMAX 数据集

Fig. 5 18 datasets from IMAX

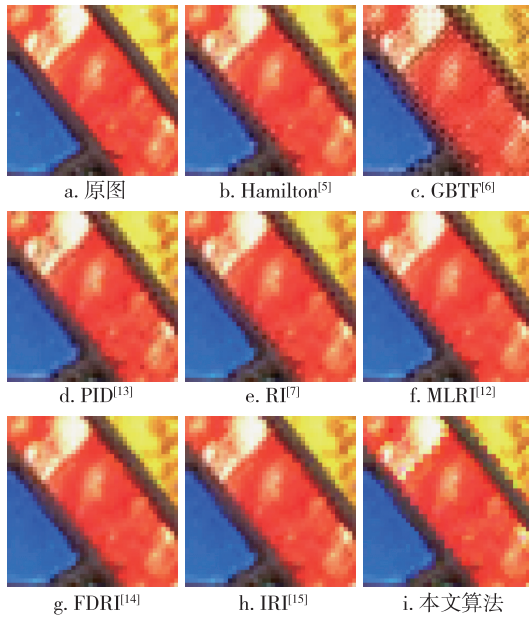


图6 各算法在IMAX数据集上的视觉比较

Fig. 6 Visual comparison of the demosaicing effect for the oblique region from the IMAX dataset

在本实验中,选择去马赛克图像和真实图像之间的峰值信噪比(PSNR)作为评价指标.PSNR值越大说明复原的效果越好.将提出的RICNN与目前效果较好的几种算法作对比,包括Hamilton^[5]、GBTF^[6]、PID^[12]、RI^[7]、MLRI^[13]、FDRI^[14]、IRI^[15]、LSSC^[16]等算法.表1和表2展示了各方法在IMAX 18数据集上的CPSNR和运行的平均时间结果对比,表1中结果最好的算法用黑色加粗字体标出.可以看出,本文方法在IMAX数据集上的峰值信噪比高达37.14 dB,比IRI和LSSC分别高出0.08和1.09 dB,但是IRI运行时间比本文算法多出1.7倍,而LSSC则多出了157倍,综合考虑去马赛克恢复质量和时间,本文提出的算法具有明显的优势.

3 结论

本文提出基于残余插值的卷积神经网络算法来去马赛克,通过实验证明了插值效果的提升.卷积神经网络在超分辨率问题中表现出众,基于这个观察,考虑将其作为一个去马赛克后续的步骤来对不太完美的图像进一步提升,实验证明,这种提升效果显著.另外还对残余插值进行了改进,一般的去马赛克算法都只在水平和垂直方向检测边缘,本文则考虑了更多方向,从而在斜边上取得了优于现有算法的效果.结果显示本文提出的算法在去马赛克问题上效果显著,接下来的工作将尝试采用其他的基于学习的方法来对图像进行去马赛克.

表1 各算法在IMAX数据集上的平均峰值信噪比(CPSNR)

Table 1 Comparison of PSNR performance for the 18 images from IMAX

序号	Hamilton ^[5]	GBTF ^[6]	PID ^[12]	RI ^[7]	MLRI ^[13]	IRI ^[15]	FDRI ^[14]	LSSC ^[16]	本文算法
1	28.12	26.95	29.08	29.12	28.97	29.12	29.38	29.06	29.61
2	33.92	33.80	34.62	35.00	35.08	35.03	35.16	34.64	35.45
3	30.56	32.77	32.19	33.75	33.86	34.29	33.54	32.16	33.73
4	33.02	35.20	36.39	37.86	37.64	37.13	38.06	36.41	38.08
5	33.04	31.27	34.06	33.91	33.99	34.70	34.83	34.08	34.77
6	36.40	33.66	37.60	38.29	38.26	38.82	38.91	37.58	39.50
7	34.03	38.21	34.81	36.95	37.46	39.17	35.85	34.80	36.44
8	35.77	37.75	36.61	36.96	36.94	39.03	38.17	36.62	37.98
9	35.49	34.52	36.78	35.92	36.44	37.49	37.54	36.76	37.48
10	37.51	36.12	38.28	38.15	38.62	38.87	39.02	38.30	39.37
11	38.23	37.04	39.47	39.43	39.91	39.85	40.02	39.45	40.34
12	37.38	36.73	38.21	39.61	39.64	39.78	39.95	38.23	39.66
13	39.81	38.89	40.80	40.27	40.51	40.47	41.01	40.81	40.85
14	37.73	37.30	38.60	38.92	38.76	38.85	39.32	38.59	39.34
15	38.13	37.19	39.09	38.37	38.91	39.31	39.45	39.03	39.46
16	32.30	30.28	34.19	35.17	35.09	35.15	35.42	34.25	35.70
17	31.92	28.99	33.17	32.44	32.58	33.68	33.76	33.12	33.88
18	33.62	34.11	35.55	36.52	36.12	36.26	36.21	35.55	36.37
平均值	34.83	34.49	36.09	36.48	36.60	37.06	36.98	36.05	37.14

表2 各算法在IMAX数据集上的运行时间

Table 2 Comparison of average running time for the IMAX datasets

	Hamilton	GBTf	PID	RI	MLRI	IRI	FDRI	LSSC	本文算法
<i>t/s</i>	0.24	10.60	1.90	2.62	0.54	5.12	0.94	453.58	2.88

参考文献

References

- [1] Bayer B E. Color imaging array: U.S. Patent 3971065 [P]. 1976-07-20
- [2] Hirakawa K, Wolfe P J. Spatio-spectral color filter array design for optimal image recovery [J]. IEEE Transactions on Image Processing, 2008, 17 (10): 1876-1890
- [3] Condat L. A new color filter array with optimal properties for noiseless and noisy color image acquisition [J]. IEEE Transactions on Image Processing, 2011, 20 (8): 2200-2210
- [4] Menon D, Calvagno G. Color image demosaicking: An overview [J]. Signal Processing Image Communication, 2011, 26 (8): 518-533
- [5] Hamilton J F Jr, Adams J E Jr. Adaptive color plan interpolation in single sensor color electronic camera: U.S. Patent 5629734 [P]. 1997-05-13
- [6] Pekkucuksen I, Altunbasak Y. Gradient based threshold free color filter array interpolation [C] // IEEE International Conference on Image Processing, 2010: 137-140
- [7] Kiku D, Monno Y, Tanaka M, et al. Residual interpolation for color image demosaicking [C] // IEEE International Conference on Image Processing, 2013: 2304-2308
- [8] He K M, Sun J, Tang X O. Guided image filtering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35 (6): 1397-1409
- [9] Dong C, Loy C C, He K M, et al. Image super-resolution using deep convolutional networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38 (2): 295-307
- [10] Timofte R, De Smet V, Van Gool L. Anchored neighborhood regression for fast example-based super-resolution [C] // IEEE International Conference on Computer Vision, 2013: 1920-1927
- [11] Zhang L, Wu X L. Color demosaicking via directional linear minimum mean square-error estimation [J]. IEEE Transactions on Image Processing, 2005, 14 (12): 2167-2178
- [12] Wu J J, Anisetti M, Wu W, et al. Bayer demosaicking with polynomial interpolation [J]. IEEE Transactions on Image Processing, 2016, 25 (11): 5369-5382
- [13] Kiku D, Monno Y, Tanaka M, et al. Minimized-Laplacian residual interpolation for color image demosaicking [J]. Proceedings of SPIE, 2014, 9023 (1): 2304-2308
- [14] Kim Y, Jeong J. Four-direction residual interpolation for demosaicking [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 26 (5): 881-890
- [15] Ye W, Ma K K. Image demosaicing by using iterative residual interpolation [C] // IEEE International Conference on Image Processing, 2014: 1862-1866
- [16] Mairal J, Bach F, Ponce J, et al. Non-local sparse models for image restoration [C] // IEEE International Conference on Computer Vision, 2009: 2272-2279

Demosaicing based on residual interpolation and convolutional neural networks

JIA Huimiao¹ LI Chunping¹ ZHOU Dengwen¹

¹ School of Control and Computer Engineering, North China Electric Power University, Beijing 102206

Abstract In order to accurately restore the texture on the oblique edges and improve the overall resolution of the demosaiced image, a convolutional neural network demosaicing algorithm is proposed based on residual interpolation. The algorithm uses the information of Bayer color filter arrays to calculate the gradient of diagonal edges, which can be used to determine the edge directions. Therefore, the corresponding interpolation formula is proposed for different edges. We incorporate the convolutional neural networks into our method to refine the interpolated images. To demonstrate the superiority of the proposed algorithm, several experiments were conducted with IMAX dataset. The experimental results show that the proposed algorithm exhibits better visual effect, higher PSNR and shorter running time compared with those of commonly used Bayer demosaicing algorithms.

Key words demosaicing; Bayer CFA; residual interpolation; edge detecting; convolutional neural networks