CAC algorithm module

Revison History

Revision	Date	Author	Description
0.1	2022.8.24	Wanwei Xiao	Initial draft

1. Introduction

1.1 Request & Purpose

This document presents chromatic aberration correction algorithm and specification in CTL ISP pipeline architecture. It defines the features with high-level diagram and modules design. The team members can follow this document to do detailed design and implementation.

1.2 Definitions & Abbreviations

Name	Description
CAC	chromatic aberration correction

1.3 Reference

[1]

2. Overview

The CAC module is used to eliminate chromatic aberrations in the image generated by the lens, generally including horizontal and axial chromatic aberrations, as well as the purple fringing phenomenon of some images

Figure 2–1 Function of cac module[2]

2.1 CAC Location

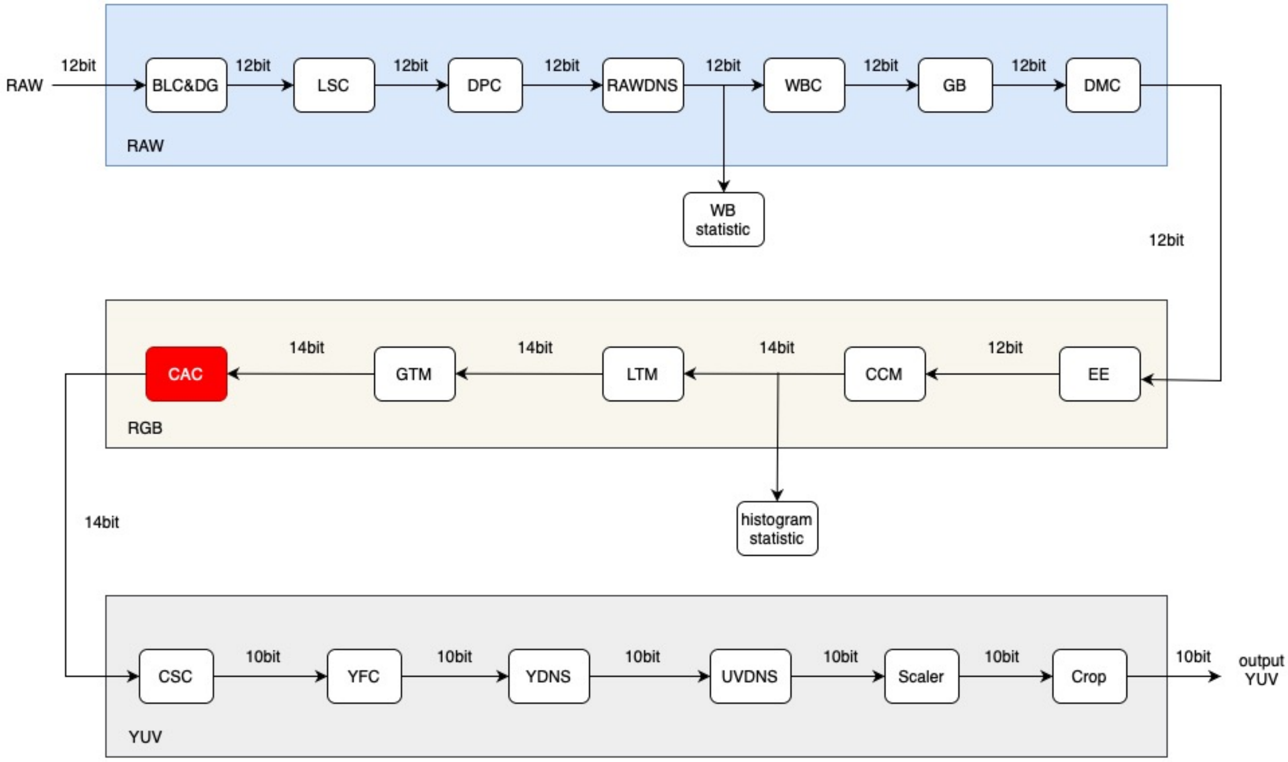


Figure 2–2 CAC location in ISP pipeline

2.2 CAC parameters

Name	Default Value	Shadow	Description
eb	1		enable signal for CAC module
t_transient	7000		determination threshold for transient region
t_edge	4000		determination threshold for edges of transient region

3. CAC algorithm and process

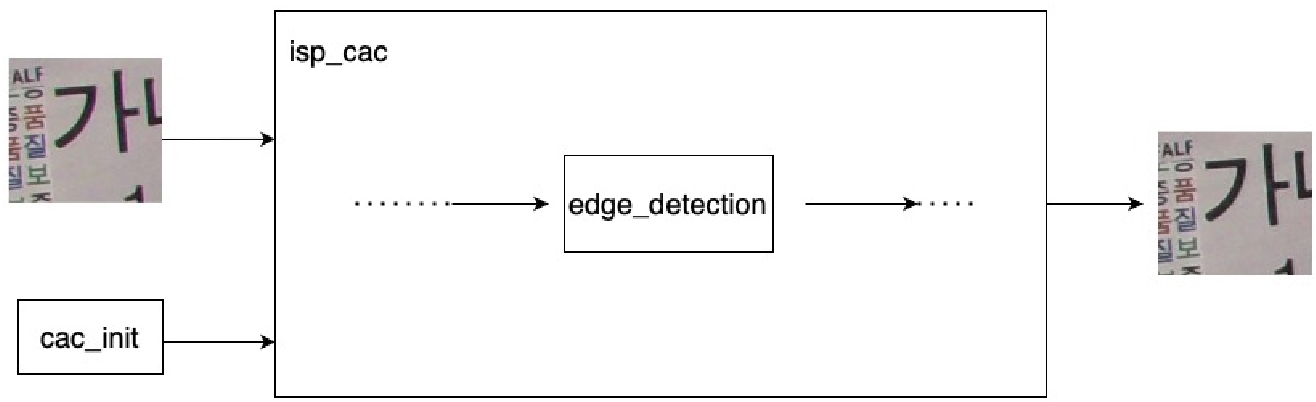


Figure 3–1 CAC module

- top parameters

Name	Description
frameWidth	the width of image
frameHeight	the height of image
inputPattern	the pattern of bayer, 0:r 1:Gr 2:Gb 3:b
blc	black pixel value

3.1 parameters initialization (cac_init)

3.1.1 function interfaces

Name	Description
top_parameter	ISP top_level parameters
cac_parameter	CAC module parameters

3.1.2 Algorithm and function

Initialize all parameter values

3.2 Cac top module (isp_cac)

3.2.1 function interfaces

Name	Description
top_parameter	ISP top_level parameters
cac_parameter	CAC module parameters
src_data	the data of input
dst_data	the data of output

3.2.2 Algorithm and function

The whole procedure can be divided into 5 stages.

Stage1: load the input image

Stage2: get the 7x7 r_window, g_window, b_window in the input pixel

Stage3: execute edge_detection function in the 7x7 windows.

Stage4: return the 7x7 windows values

Stage5: output images

3.3 CAC algorithm execution (edge_detection)

3.3.1 function interfaces

Name	Description
top_parameter	ISP top_level parameters
cac_parameter	CAC module parameters
r_window[7][7]	the processing window of r channel
g_window[7][7]	the processing window of g channel
b_window[7][7]	the processing window of b channel

3.3.2 Algorithm and function

This procedure can be divided into horizontal and vertical direction processs. Take the horizontal direction as an example.

The whole procedure can be divided into 2 stages.

Stage1:chromatic difference and edge information calculation

chromatic difference calculation

- rg_diff = inPixel_int.r – inpPixel_int.g
- bg_diff = inPixel_int.b – inPixel_int.g

chromatic difference calculation

C_edge_d = Ec(i,j) = C(i–1,j–1) + 2C(i,j–1) + C(i+1,j–1) – C(i–1,j+1) – 2C(i,j+1) – C(i+1,j+1) //C is the {R,G,B}, d represent the direction, horizontal or vertical

Stage2: transient region and the edge of transient calculation, this Stage can be divided into 2 steps.

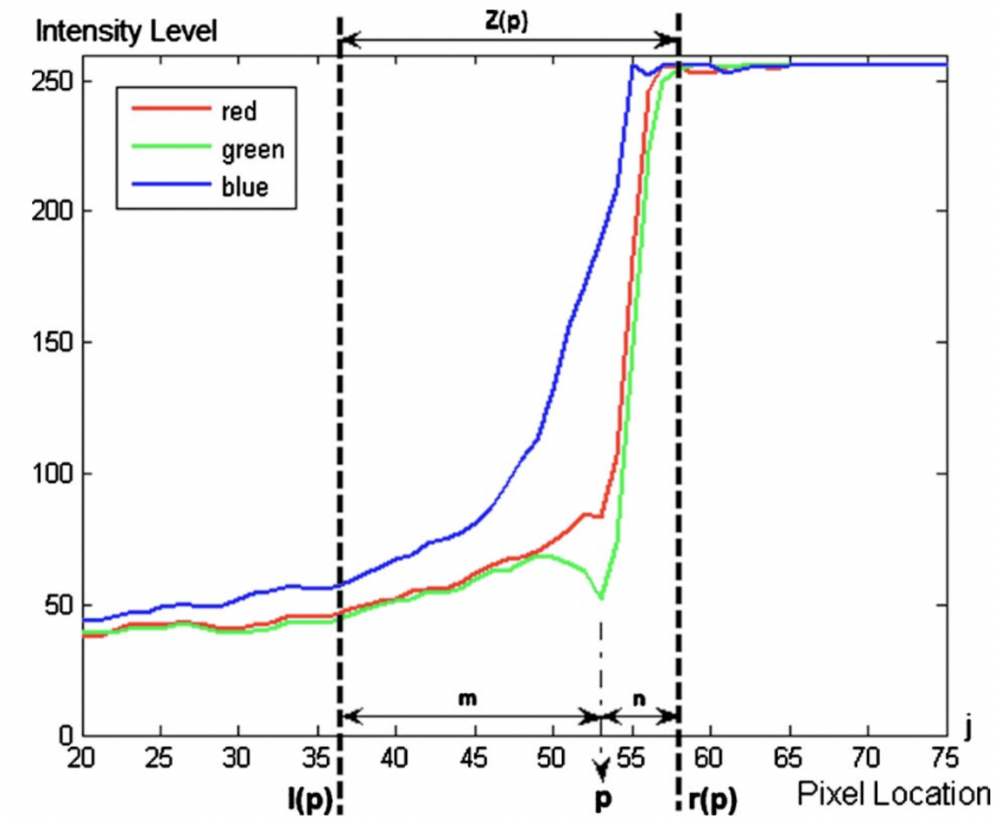


Figure3-2. the location of p, l_t(l), and r_t(r)[1]

step1:calculate the transient pixel p

j <- j+1 while abs(Eg(i,j)) < t_transient

set p = j if abs(Eg(i,j)) >=T

step2:search l_t and r_t

the sign of the gradient value of the green signal an the initial point p

sigma_temp = sgn[Eg(p)]

the largest gradient value among r,g,b at x that has similar edge to g at given initial pixel p is defined as follows:

H(x|p) = max{sigam_temp*Er(x), sigma_temp*Eg(x), sigma_temp*Eb(x)}

search to the left and right until H(l|p) < t_edge and H(r|p) < t_edge

l_t = l, r_t = r

Stage3:this step is the pixel correction, which includes all pixel values from l_t to r_t, with l_t and r_t as reference pixels. The main constraint of the correction process is as follows.

- min{rg_diff[l_t], rg_diff[r_t]} <= rg_diff[i] <= max{rg_diff[l_t], rg_diff[r_t]}   //i is the pixel between l_t and r_t

- min{bg_diff[l_t], bg_diff[r_t]} <= bg_diff[i] <= max{bg_diff[l_t], bg_diff[r_t]} //i is the pixel between l_t and r_t

so the correction equation is as follows.

$$\hat{R}(j)=\begin{cases}\min\{D_R[l(p)],D_R[r(p)]\}+G(j), & \text{if } D_R(j)<\min\{D_R[l(p)],D_R[r(p)]\}\\ \max\{D_R[l(p)],D_R[r(p)]\}+G(j), & \text{if } D_R(j)>\max\{D_R[l(p)],D_R[r(p)]\},\\ R(j), & \text{otherwise}\end{cases}$$

$$\hat{B}(j)=\begin{cases}\min\{D_B[l(p)],D_B[r(p)]\}+G(j), & \text{if } D_B(j)<\min\{D_B[l(p)],D_B[r(p)]\}\\ \max\{D_B[l(p)],D_B[r(p)]\}+G(j), & \text{if } D_B(j)>\max\{D_B[l(p)],D_B[r(p)]\},\\ B(j), & \text{otherwise}\end{cases}$$

Figure 3–3 is a correction example, and the l_t = 1, r_t = 4, and the correction effect is as shown in figure 3–4

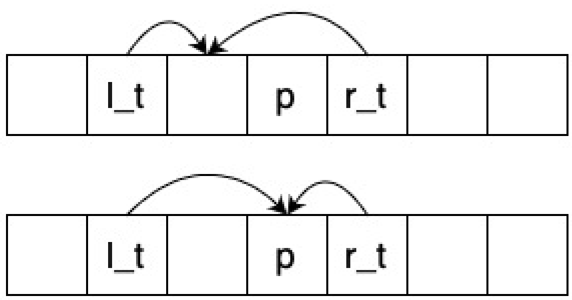
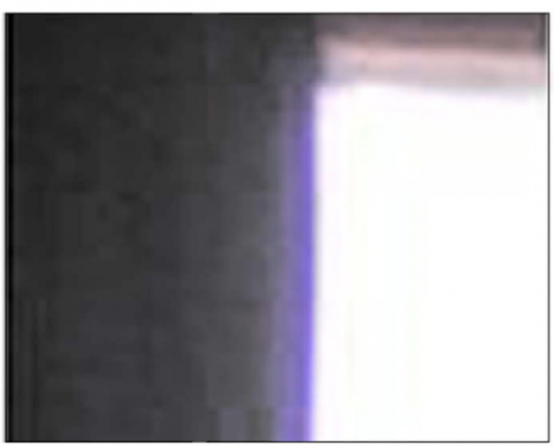
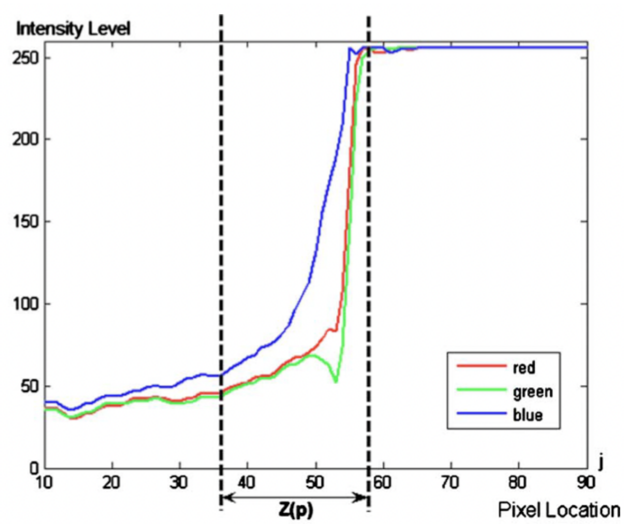


Figure 3–3. correction example





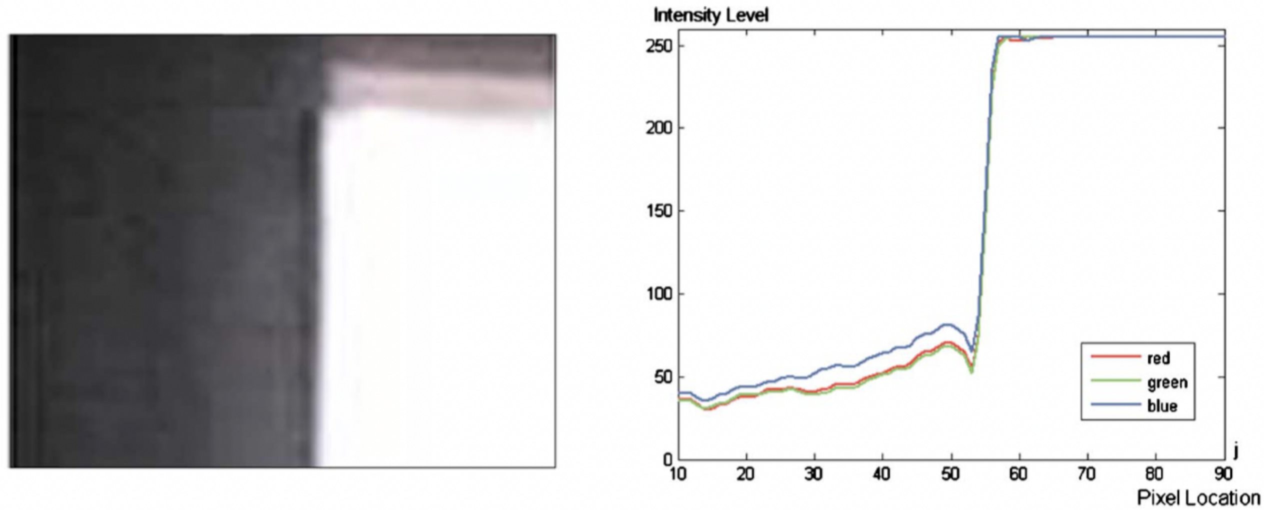


Figure3-4. correction effect and rgb intensity[1]

4. Reference

[1] Chung S W , Kim B K , Song W J . Removing chromatic aberration by digital image processing[J]. Optical Engineering, 2010, 49.

[2] Chang J , Kang H , Kang M G . Correction of Axial and Lateral Chromatic Aberration With False Color Filtering[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2013, 22(3):1186-1198.