ASIC implementation of automatic gamma correction based on average of brightness

Document of Bachelor Thesis

Instructor:

Dr. Mahdiani

Student:

Pedram Babakhani

Pe.Babakhani@sbu.ac.ir

Department of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran

Abstract

Preprocessing is essential stage in image processing because of limitation of imaging device or inappropriate environmental light. This paper presents a preprocessing technique for estimating the amount of gamma correction in the absence of any information or knowledge about environmental light and imaging device. The basic approach exploits the amount of gamma correction based on average brightness. The amount of gamma correction is then estimated by a power which transports average of brightness to center of histogram.

Keywords: *average of brightness, preprocessing, histogram, gamma Correction, execution time*

1. Introduction

Luminance is an important factor in image processing that leads to perception of details. Technical limitation of imaging devices results non-linear effects on image. Gamma correction is non-linear operation which enhances brightness of image. Gamma correction is defined by the following power law expression:

$$S = T(R) = R^{\gamma}$$

S is the value of brightness in output image and R is value of brightness in original image that are mapped to [0 1].

If the value of gamma is known, then inverting this process is obvious:

$$S = T(r) = r^{\frac{1}{\gamma}}$$

Gamma correction would be advantageous to remove non-linear effects in preprocessing stage for many applications in digital photography, image processing, and computer vision. In this paper a

technique is presented for estimating the amount of gamma correction in the absence of any information or knowledge about environmental light or imaging device. The basic approach exploits the fact that amount of gamma correction is determined by transposing average of brightness to center of histogram.

2. Proposed method

Average of brightness is simple element that can be computed in the least amount of time. Basic approach in this article present a technique to estimate appropriate gamma based on average brightness. Although average of brightness doesn’t present all information about image, it is the best choice to choose a sample between amounts of brightness in histogram. This method presents a technique which is different method and low order to estimate gamma. This paper proposes a method which estimates a power that transport average amount of brightness to center of histogram. This method extends the estimated power for gamma correction. This power can be chosen as global gamma for gamma correction. We suppose a gamma which changes average of brightness to $\frac{1}{2}$, then gamma is estimated based on following equations:

$$X^{\gamma} = \frac{1}{2}$$

$$\gamma = \log_{X} \frac{1}{2}$$

$$\gamma = \frac{\log_{10} \frac{1}{2}}{\log_{10} X}$$

$$\gamma = \frac{-0.3}{\log_{10} X}$$

X is average brightness and X ∈ [0 1]. In the equations, $\frac{1}{2}$ is center of histogram brighness which is global for any format and it is'nt limmited to Unint8 and int8 etc.

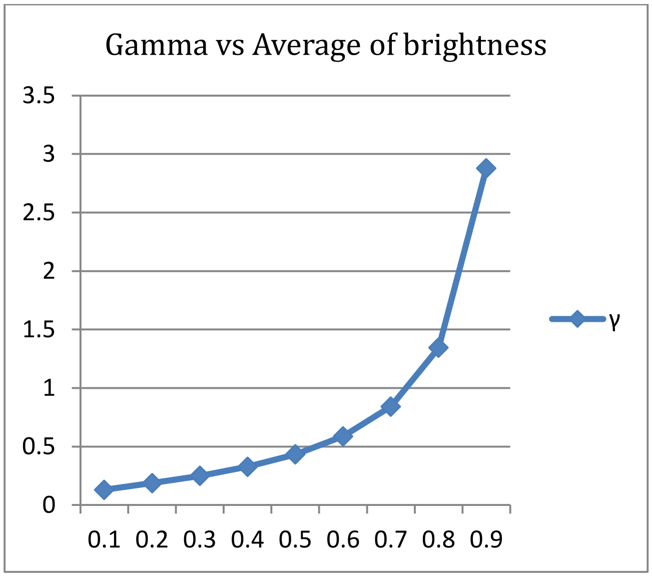


Fig. 1 Graph of Gamma Vs Average of brightness

Finally, average of brightness in output image is not $\frac{1}{2}$ because this method just chooses the average of brightness in original image as a sample to estimate gamma. All pixels in output image will be enhanced with estimated gamma. After all, input image will be enhanced with this method. Graph 1 demonstrates that proposed method estimates a logical and appropriate value for gamma correction.

3. Experimental results

In this paper we present a new preprocessing technique for estimating the gamma values without any information or knowledge of the imaging device or environmental luminance. We consider subjective and objective image quality assessment to demonstrate the performance of the proposed method. These figures are benchmark images with a high contrast and low contrast. The enhanced images bring out much more details of the original images. Quality of enhanced images indicates that the enhancement results using the proposed method have an appropriate performance compared to the other methods. In fact, our proposed method estimates gamma in least amount of time between existing approaches. Minimum execution time of proposed method is noticeable feature against other methods and algorithms.

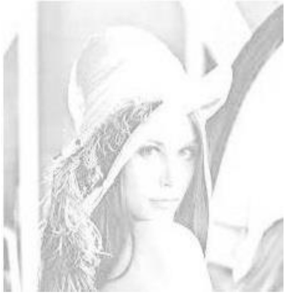


Fig. 2 original image

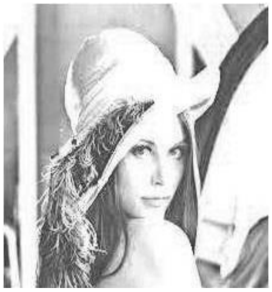


Fig. 3 enhanced image

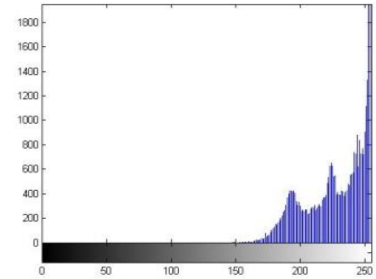


Fig. 4 original histogram

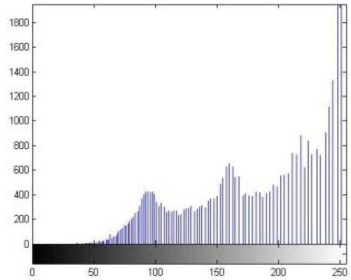


Fig. 5 enhanced histogram

Average of brightness in Figure 2 is 0.92 and estimated gamma is 3.5. more quality and more details (face and hat) in output image demonstrates that proposed method leads to enhancement.

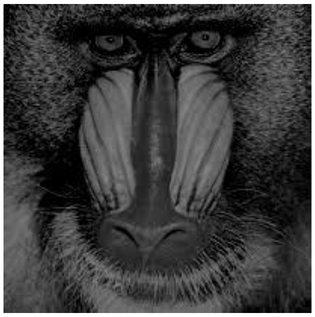


Fig. 6 original image

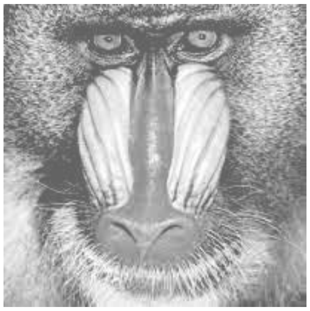


Fig. 7 enhanced image

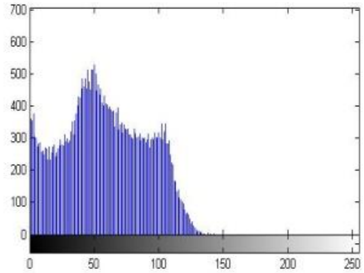


Fig. 8 original histogram

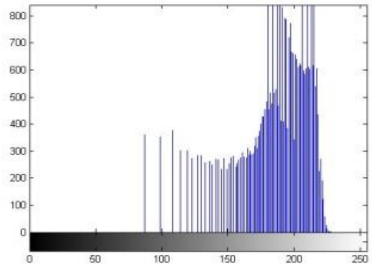


Fig. 9 enhanced histogram

Average of brightness in Figure 6 is 0.21 and estimated gamma is 0.2. more quality and more details (hair hand nose of baboon) proves that proposed method performs well.

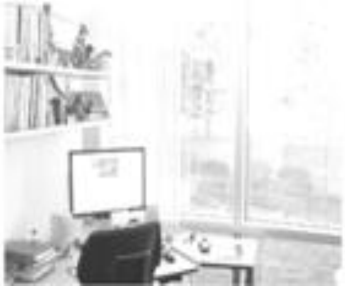


Fig. 10 original image

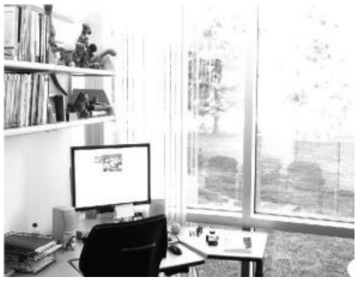


Fig. 11 enhanced image

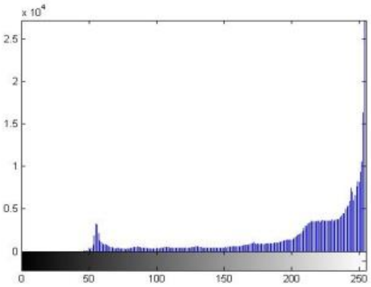


Fig. 12 original image

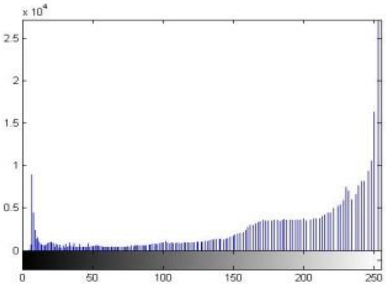


Fig. 13 enhanced image

Average of brightness in Figure 10 is 0.87 and estimated gamma is 2.33. more quality and more perceiving of details (reflection of object on mirror and details in shelf) shows that proposed method enhances original image.

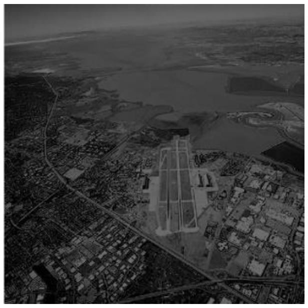


Fig. 14 original image

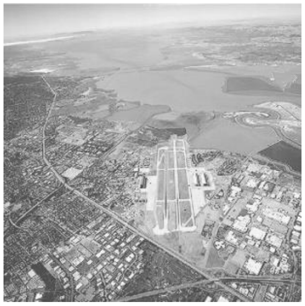


Fig. 15 enhanced image

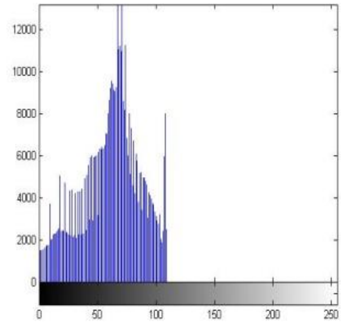


Fig. 16 original histogram

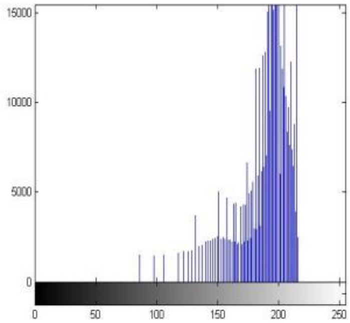


Fig. 17 enhanced histogram

Average of brightness in Figure 14 is 0.2.17 and estimated gamma is 0.19. more perceiving of details (houses and road) demonstrates that proposed method enhances image.
Histogram of all images became equalized and include more amount of brightness which leads to more contrast.

4. Comparison of results

Execution time is very important parameter in improvement of a method. Simulations demonstrate that Other methods such as local gamma correction and blind inverse gamma correction perform in more time than proposed method. All execution times are existed in Table 1.

Table 1 Execution times in different methods

Method	***Blind inverse gamma correction***	***Local gamma correction***	***Proposed method***
Execution time on 512X512	0.72 Sec	0.63 Sec	0.22 Sec
Execution time on 256X256	0.63 Sec	0.56 Sec	0.19 Sec
Execution time on 128X128	0.58 Sec	0.52 Sec	0.17 Sec
Execution time on 64X64	0.56 Sec	0.50 Sec	0.16 Sec
average	0.6225 Sec	0.5525 Sec	0.185 Sec

Execution time in proposed method is the least amount of time. Difference between proposed method and other methods is noticeable. Comparison between three mentioned methods are existed is Graph 1 and 2.

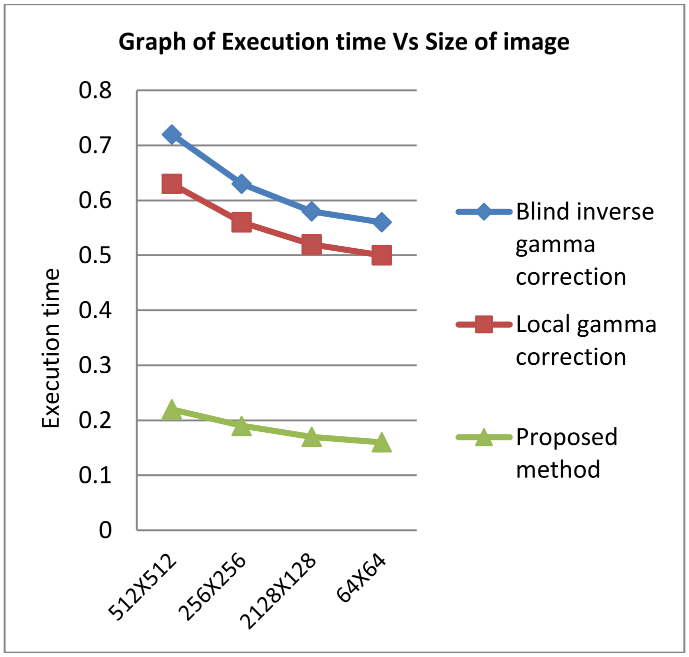


Fig. 16 Graph of Execution Time Vs Size of image

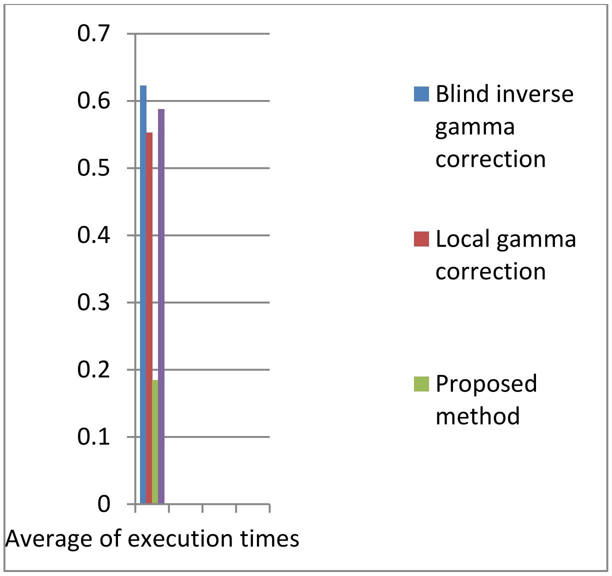


Fig. 17 Graph Average execution time Vs Methods

Execution time is reduced from average 0.5875 second to 0.185 second (68.6 %) by proposed method. In sum up, proposed method performs at least time against other methods.

5. Implementation on hardware

Latency is one of the most important parameter in implementation on hardware. We divide original image to blocks with size of 16X16. gamma estimator module selects one pixel from blocks and make a matrix then estimates gamma based on proposed method and enhanced brightness of pixels. In this method we reduce latency because we estimate gamma from smaller matrix in lower time and enhanced them with estimated gamma and drive them to output while inputs are coming. Hence we reduce latency to 1/256 because gamma estimator and gamma corrector are parallel. Our experiences indicate that estimated gamma from smaller matrix's is about global estimated gamma. Figure 18 indicates block diagram of proposed architecture.

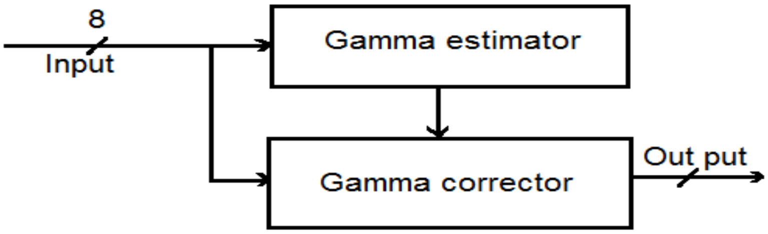


Fig. 18 Block diagram of architecture

On the following images we demonstrate our idea and prove that with estimated gamma from smaller matrix. Graph of gamma demonstrate that gamma change approximately linear with worthless alterations. (all following images are simulated with MATLAB).

```
I= imread('1.jpg');
I= rgb2gray(I);
I=mat2gray(I);
a=zeros(1,65536);
gamma= zeros(256,1);
averageofbrightness_I = (sum(sum(I))/ (256*256));
gamma_I = -0.3 / log(averageofbrightness_I);
t=1;
for m=1:1:16
for n=1:1:16
for i=n:16:256
for j=m:16:256
a(1,t) = I(i,j);
t = t+1;
end
end
end
end

x=1;
sum = 0;
for s=1:256:65536
for q=0:1:255
sum = sum + a(1,s+q);
averageofbrightness_a = sum/ (256);
gamma_a = -0.3 / log(averageofbrightness_a);
gamma(x,1) = gamma_a;
end
x = x+1;
sum = 0;
end

max = max(gamma);
min = min(gamma);
max_min = max-min;

max_error = (max - gamma_I)/ gamma_I;
min_error = (gamma_I - min) / gamma_I;

sum_of_gamma =0;
for p=1:1:256
 sum_of_gamma = sum_of_gamma + gamma(p,1);
end


average_of_gamma=sum_of_gamma /256;
```

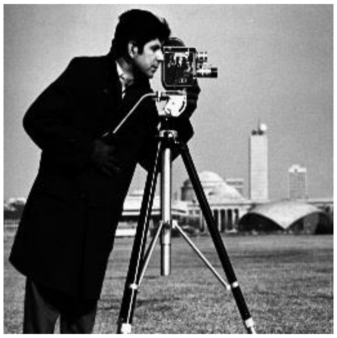


Fig. 19 original image

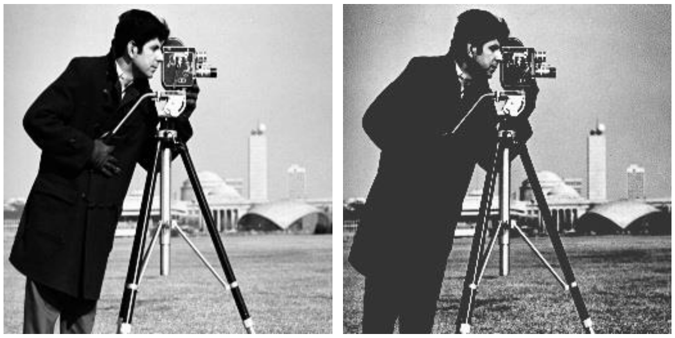


Fig. 20 enhanced image

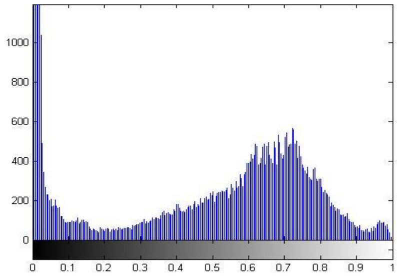


Fig. 21 original histogram

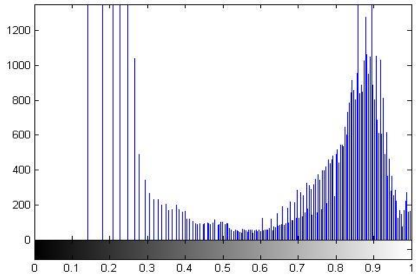


Fig. 22 enhanced histogram

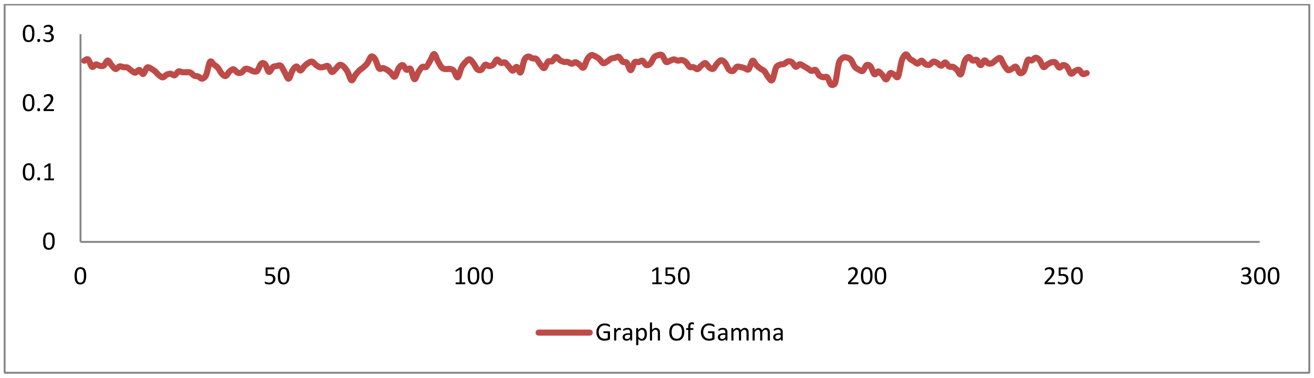
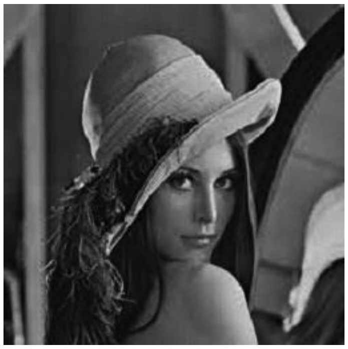


Fig. 23 Alterations of Gamma (cameraman)

Table 2 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
0.253239	0.25312	0.04463	0.07246	0.10389



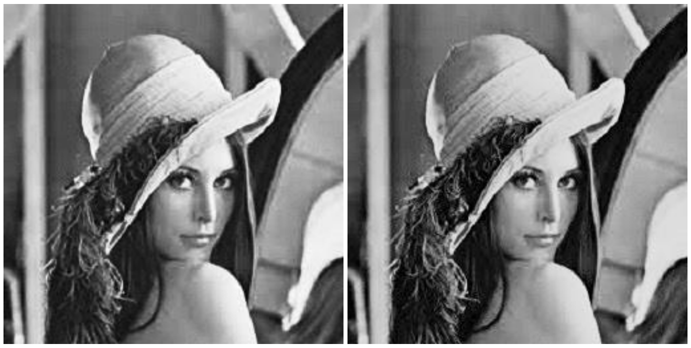
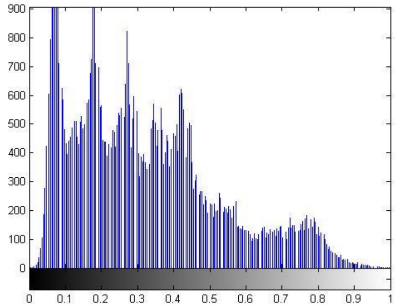


Fig. 24 original image

Fig. 25 enhanced image



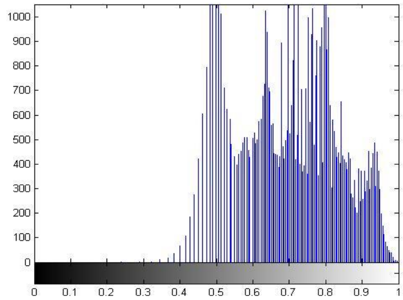


Fig. 26 original histogram

Fig. 27 enhanced histogram

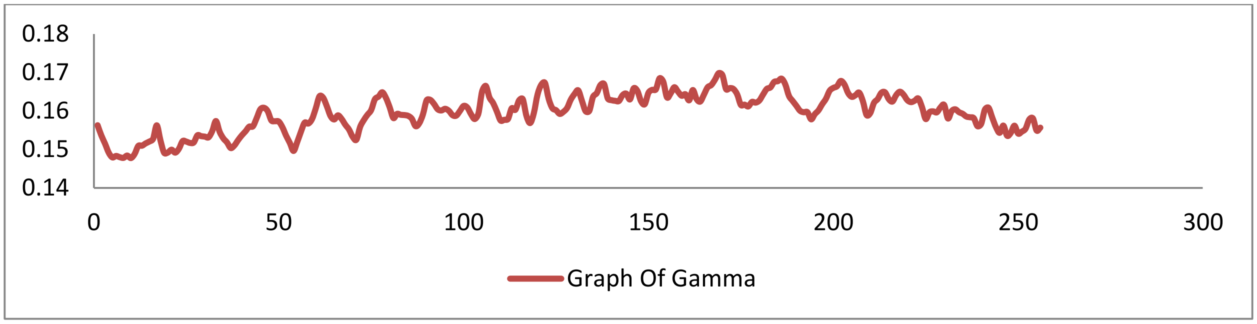


Fig. 28 Alterations of Gamma (lenna)

Table 3 Different Gamma and Error

Global gamma	***Average of gamma (1,2,…,256)***	***Max gamma – min gamma***	***Error of max= Max gamma –global gamma/ global gamma***	***Error of min= Mingamma – global gamma/ global gamma***
0.15957	0.15958	0.02208	0.06403	0.07436

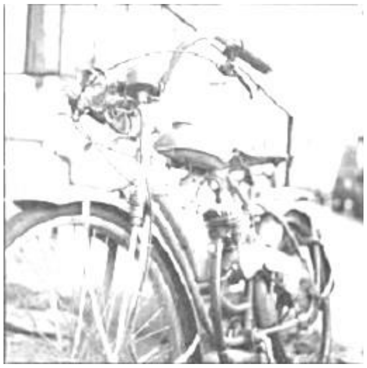


Fig. 29 original image

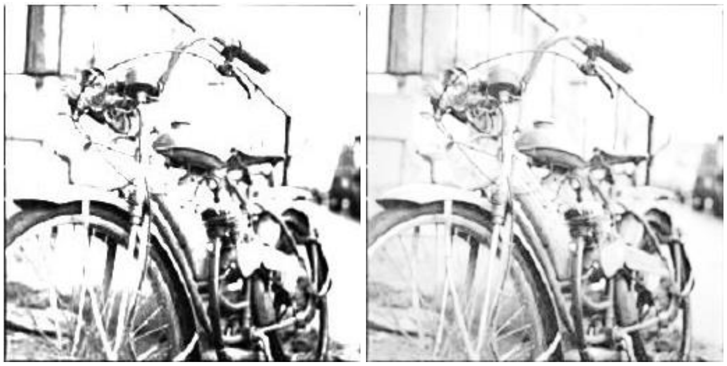


Fig. 30 enhanced image

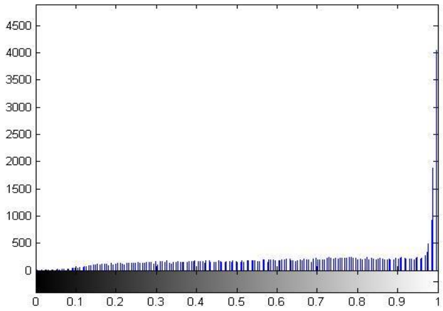


Fig. 31 original histogram

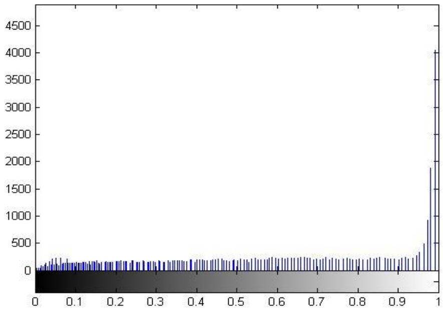


Fig. 32 enhanced histogram

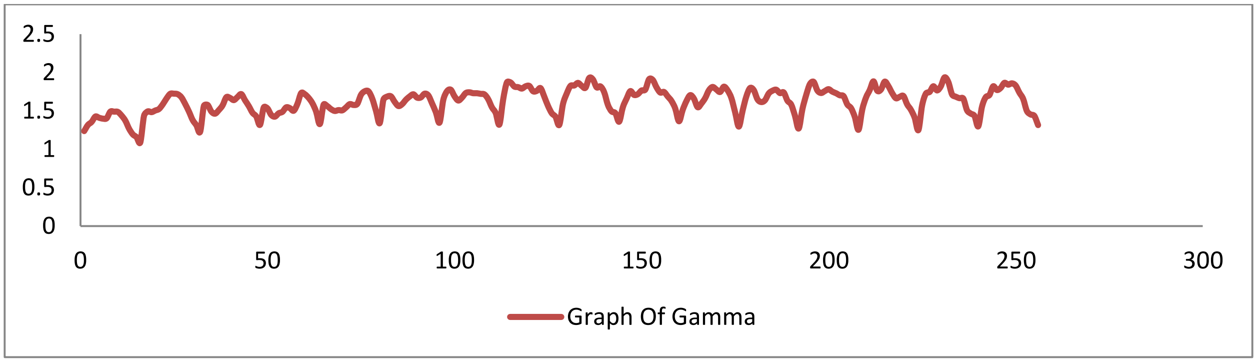


Fig. 33 Alterations of Gamma (motor bicycle)

Table 4 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
1.60441	1.62094	0.84789	0.20676	0.32170

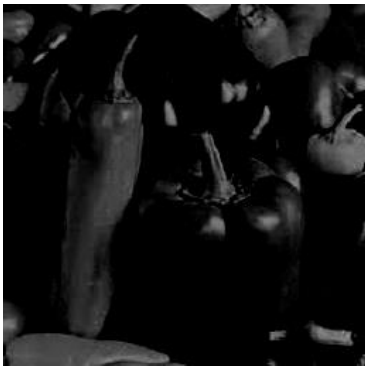


Fig. 34 original image

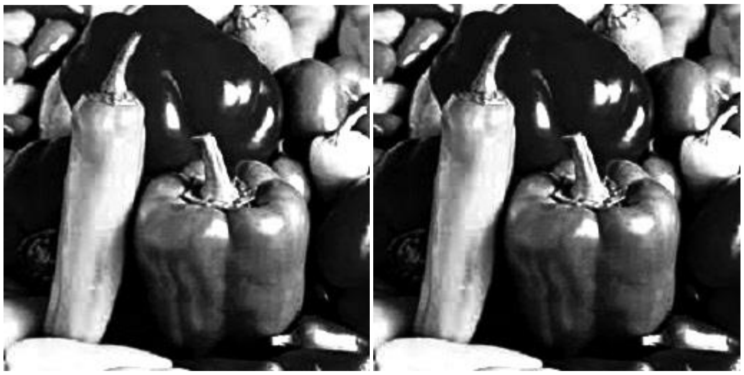


Fig. 35 enhanced image

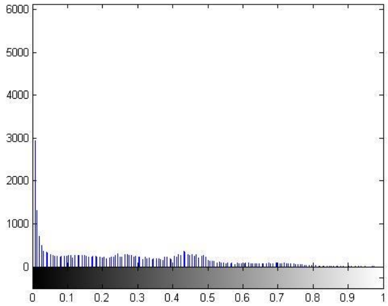


Fig. 36 original histogram

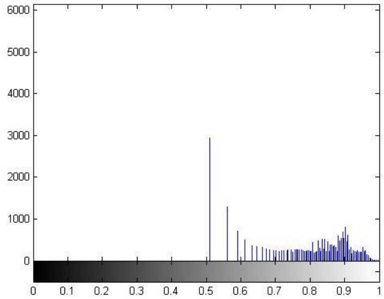


Fig. 37 enhanced histogram

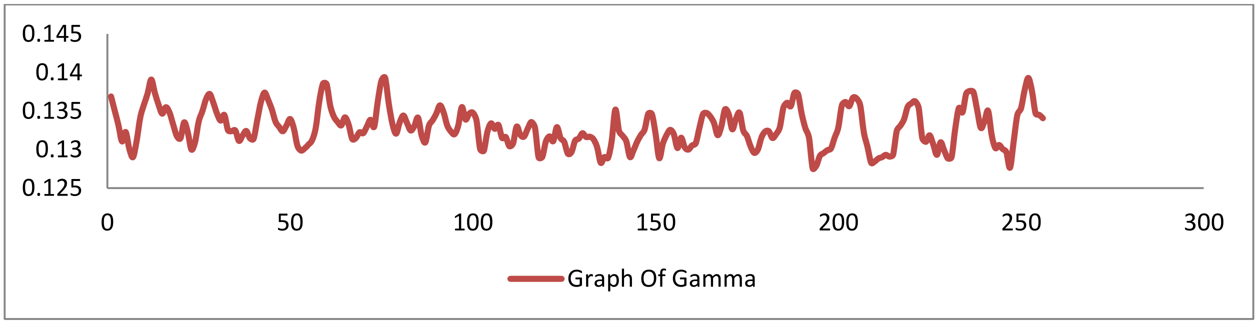


Fig. 38 Alterations of Gamma (papers)

Table 5 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
0.13283	0.13283	0.01174	0.04862	0.03981

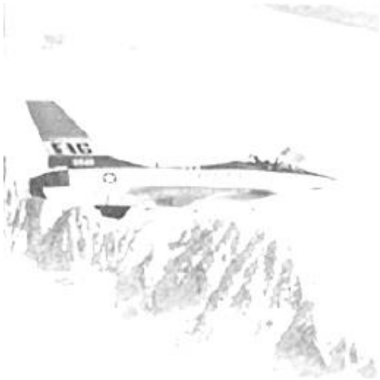


Fig. 39 original image

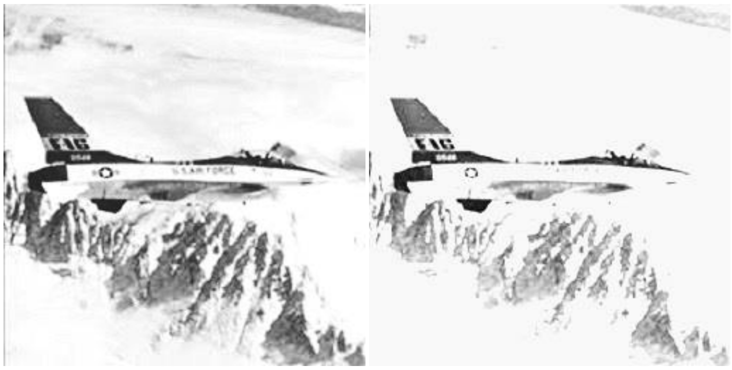


Fig. 40 enhanced image

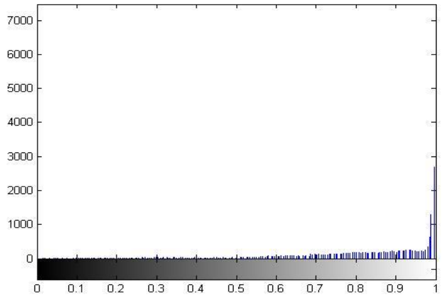


Fig. 41 original histogram

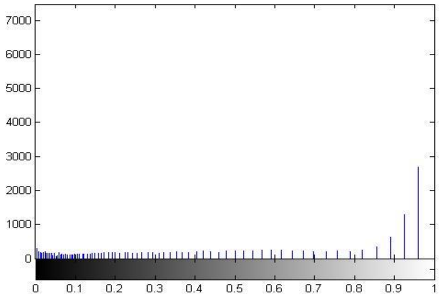


Fig. 42 enhanced histogram

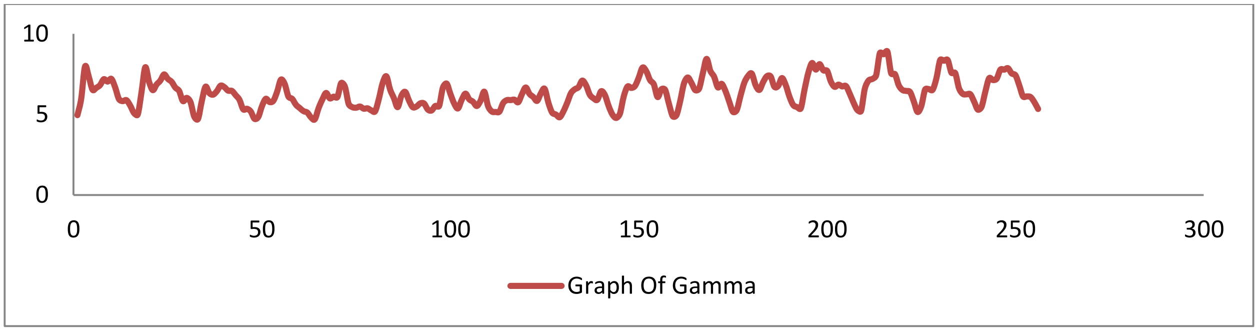


Fig. 43 Alterations of Gamma (phantom)

Table 6 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
6.22362	6.34131	4.21197	0.43157	0.24519

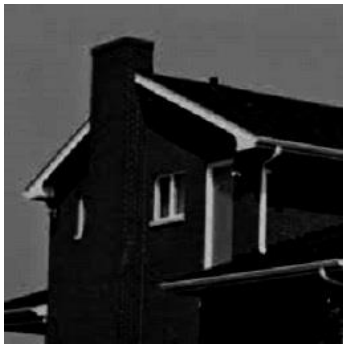


Fig. 44 original image

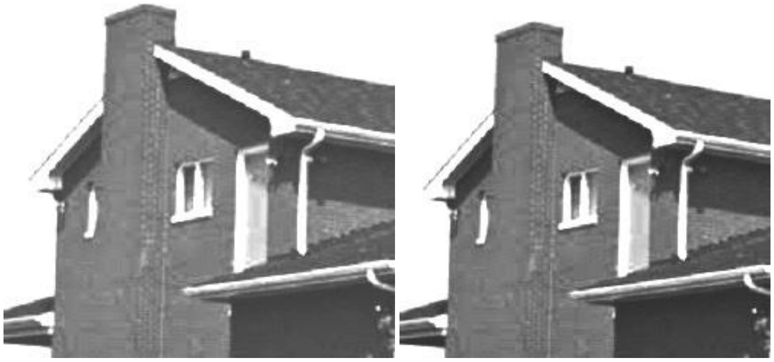


Fig. 45 enhanced image

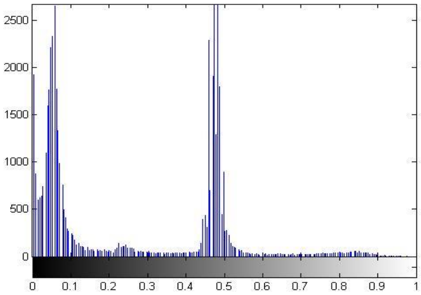


Fig. 45 original histogram

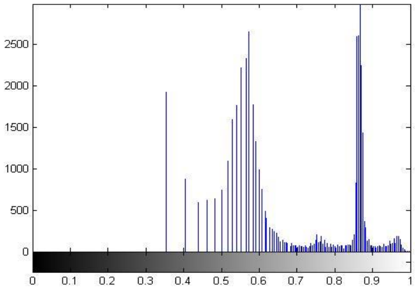


Fig. 46 enhanced histogram

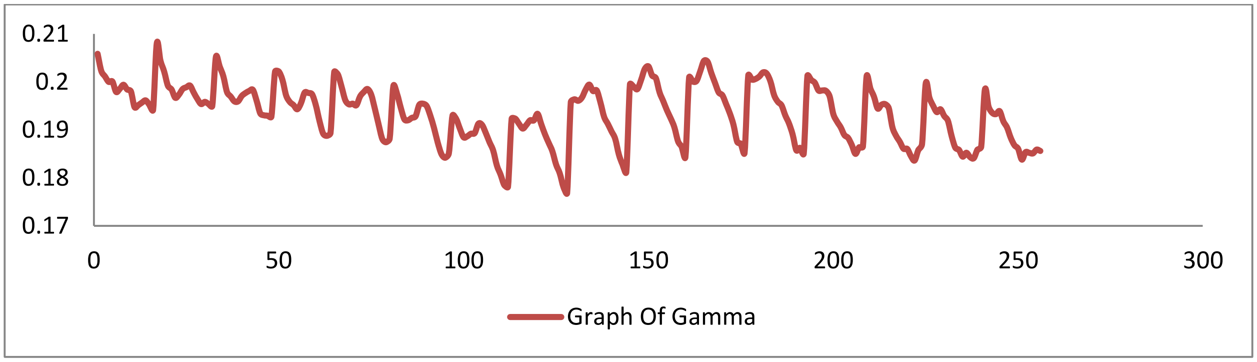


Fig. 47 Alterations of Gamma (house)

Table 7 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
0.19348	0.19352	0.03115	0.07503	0.08599

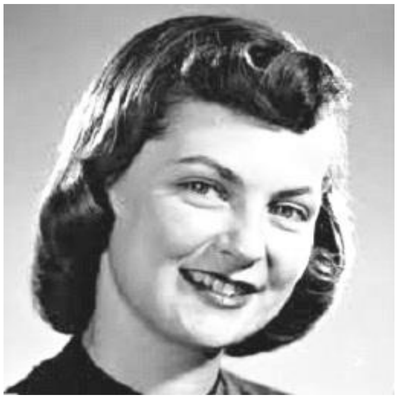


Fig. 48 original image

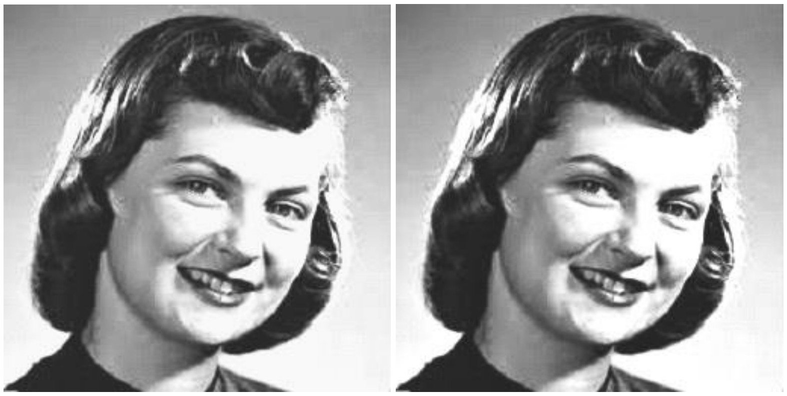


Fig. 49 enhanced image

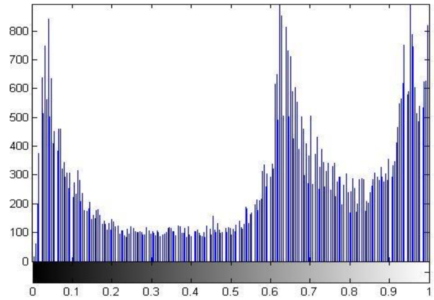


Fig. 50 original histogram

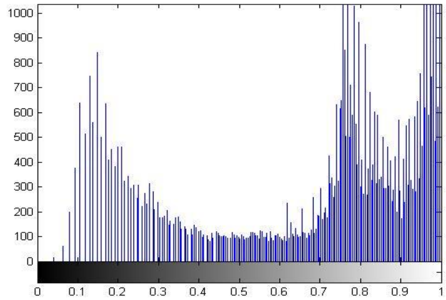


Fig. 51 enhanced histogram

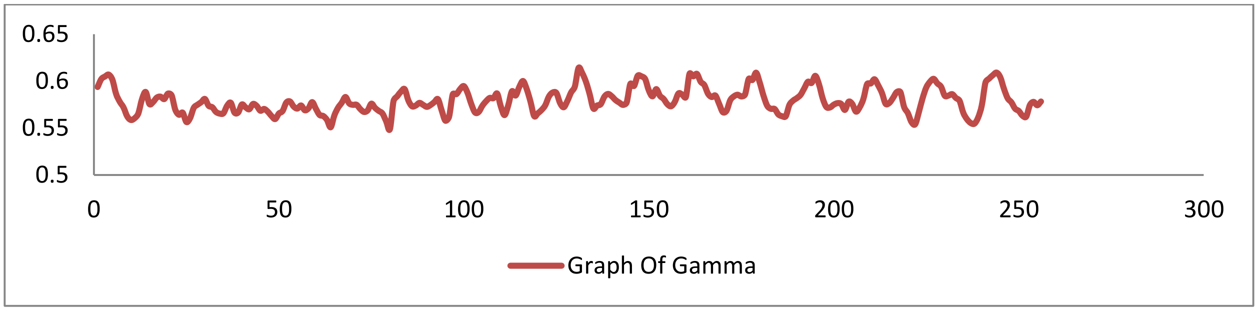


Fig. 52 Alterations of Gamma (face)

Table 8 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
0.57915	0.57936	0.06532	0.06035	0.05244

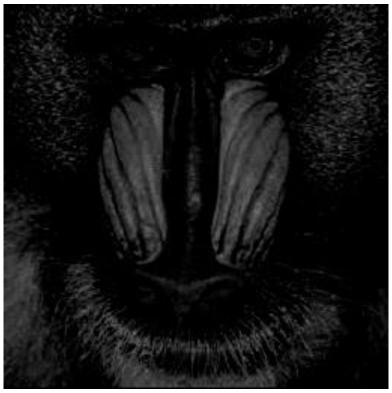


Fig. 53 original image

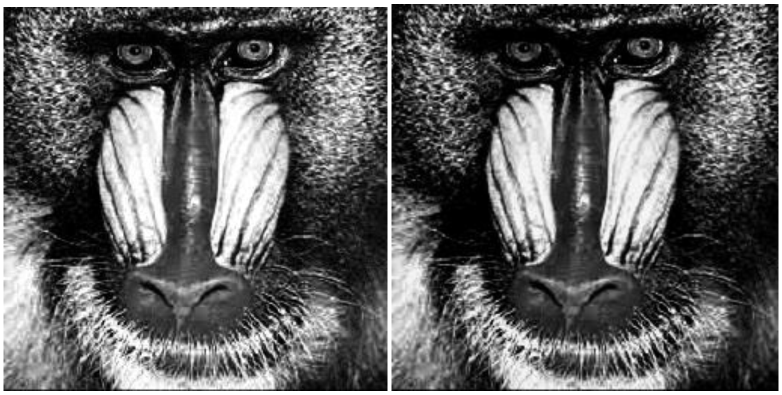


Fig. 54 enhanced image

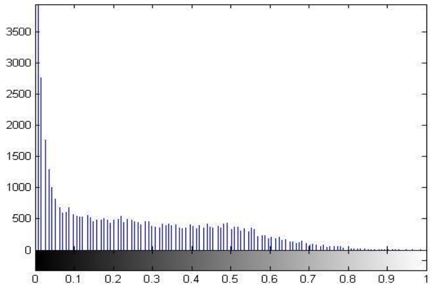


Fig. 55 original histogram

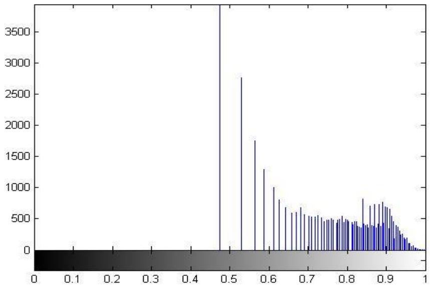


Fig. 56 enhanced histogram

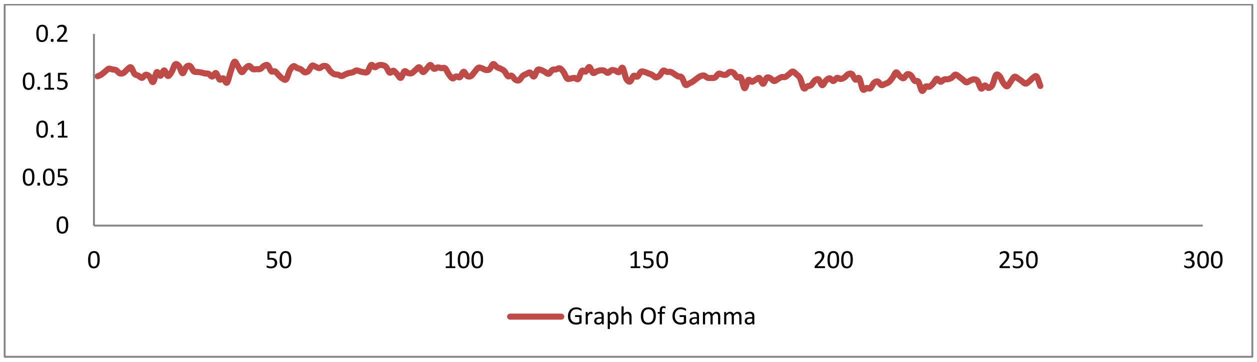


Fig. 57 Alterations of Gamma (baboon)

Table 9 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
0.15719	0.15720	0.03061	0.08900	0.10574

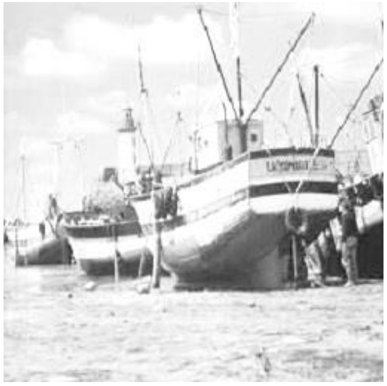


Fig. 58 original image

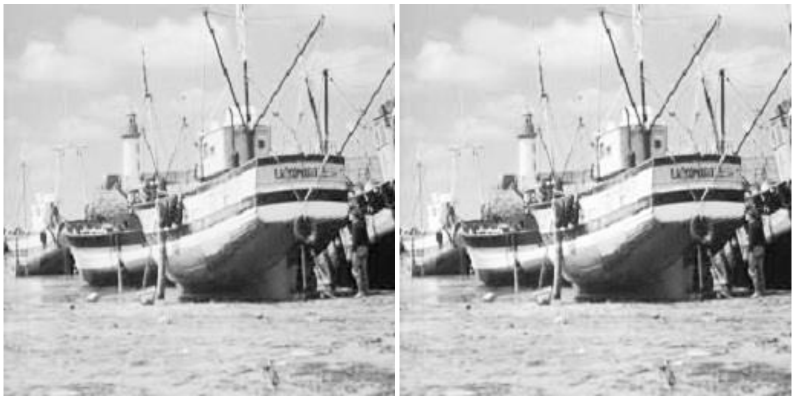


Fig. 59 enhanced image

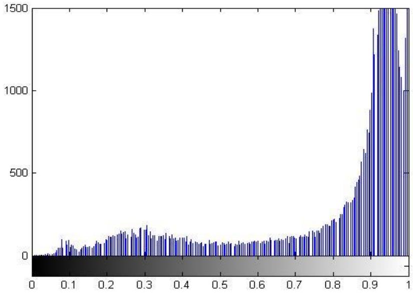


Fig. 60 original histogram

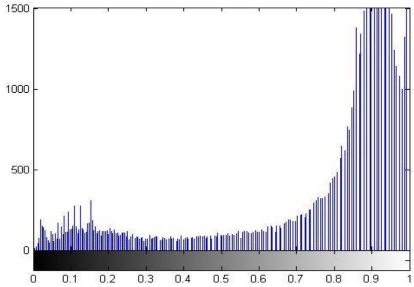


Fig. 61 enhanced histogram

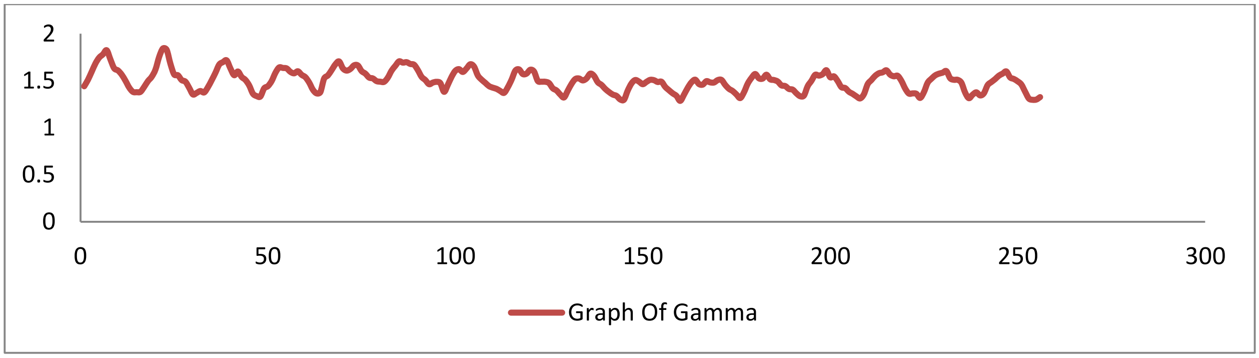


Fig. 62 Alterations of Gamma (boat)

Table 10 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
1.49331	1.50063	0.55913	0.23495	0.13947

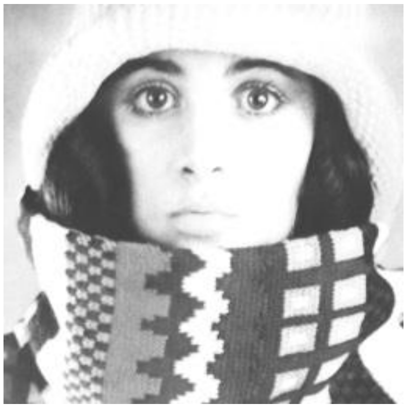


Fig. 63 original image

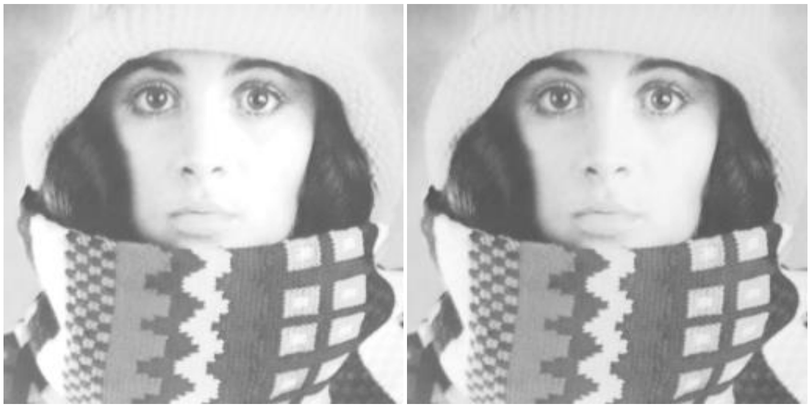


Fig. 64 enhanced image

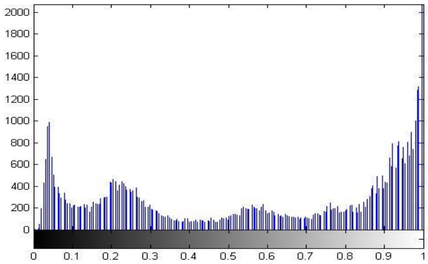


Fig. 65 original histogram

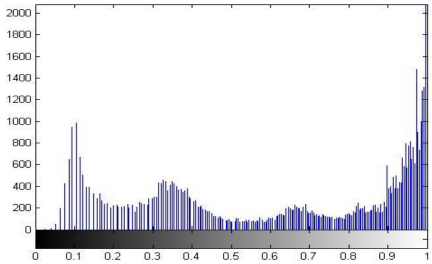


Fig. 66 enhanced histogram

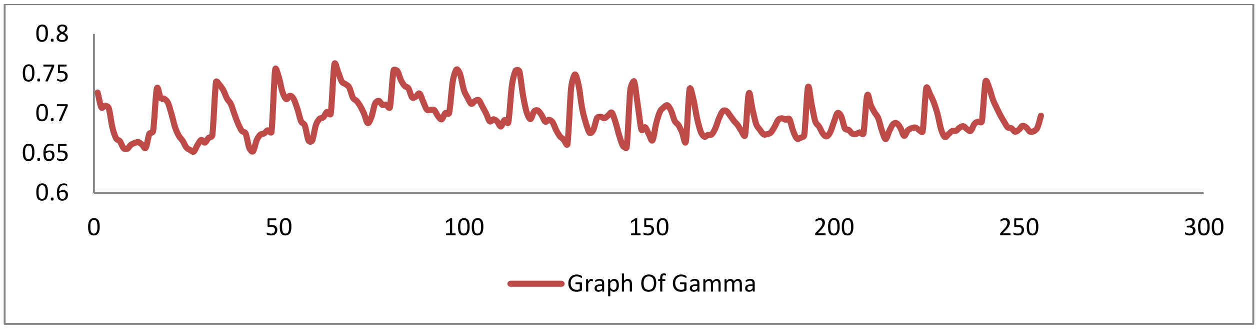


Fig. 67 Alterations of Gamma (face)

Table 11 Different Gamma and Error

Global gamma	*Average of gamma (1,2,…,256)*	*Max gamma – min gamma*	*Error of max= Max gamma –global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
0.69451	0.69518	0.10990	0.09684	0.06140

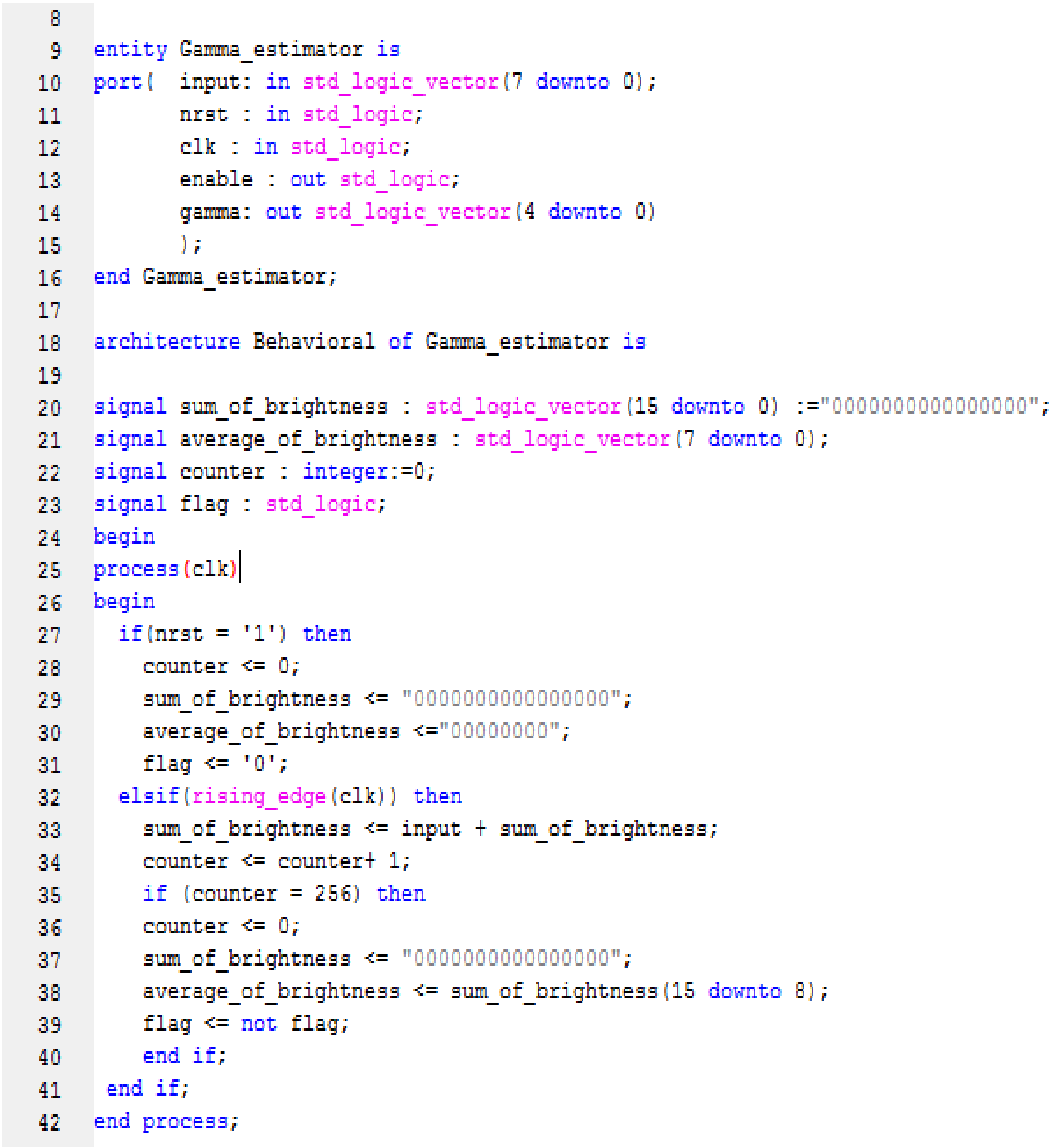
Table 12 Different Gamma and Error and Average of Errors

Image	*Global gamma*	*Average of gamma (1,2,…,n)*	*Max gamma – min gamma*	*Error of max= Max gamma – global gamma/ global gamma*	*Error of min= Mingamma – global gamma/ global gamma*
Cameraman	0.253	0.253	0.044	0.07246	0.10389
Lenna	0.159	0.159	0.022	0.06403	0.07436
Motor bicycle	1.620	1.604	0.847	0.20676	0.32170
Pepper	0.132	0.132	0.0117	0.04862	0.03981
Phantom	6.223	6.341	4.211	0.43157	0.24519
House	0.193	0.193	0.031	0.07503	0.08599
Face	0.579	0.579	0.065	0.06035	0.05244
Baboon	0.157	0.157	0.030	0.08900	0.10574
Boat	1.493	1.5006	0.559	0.23495	0.13947
Face	0.695	0.694	0.109	0.09684	0.06140
Average of Errors				0.137961	0.12299

Average of errors in examples and tested images is about %13. All images are different such as high frequency and low frequency. This error demonstrates that all output images which are based on proposed method and proposed implementation are variances from exact output image. In international scientific view, %13 can be reliable. We can extract from table 12 that error is lower in dark images and is higher in light images. All in all, we can use this method in lower execution time and implement it on hardware in parallel processing between gamma estimator and gamma correction module.

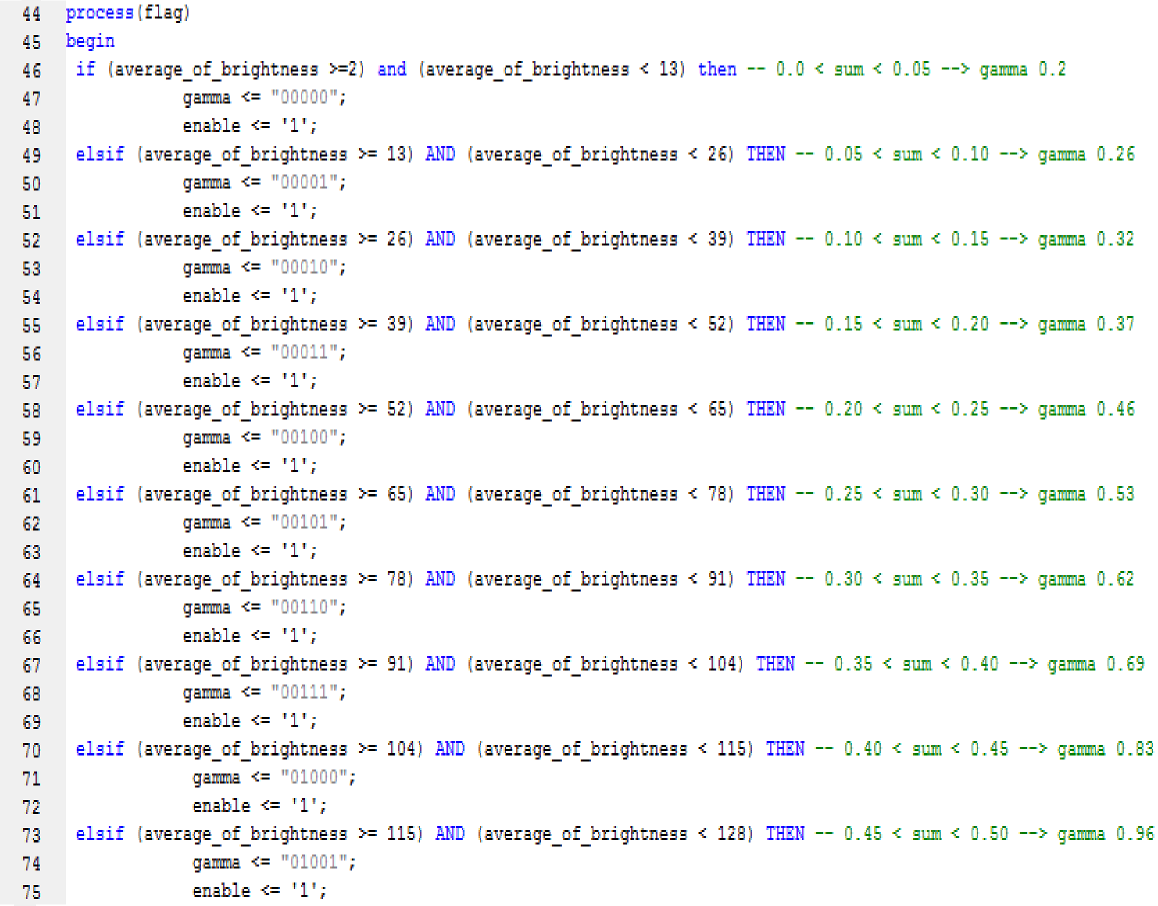
5.1. Gamma estimator module

This module gets 256 inputs (16x16) then estimates gamma based on average brightness and send gamma to gamma corrector module.

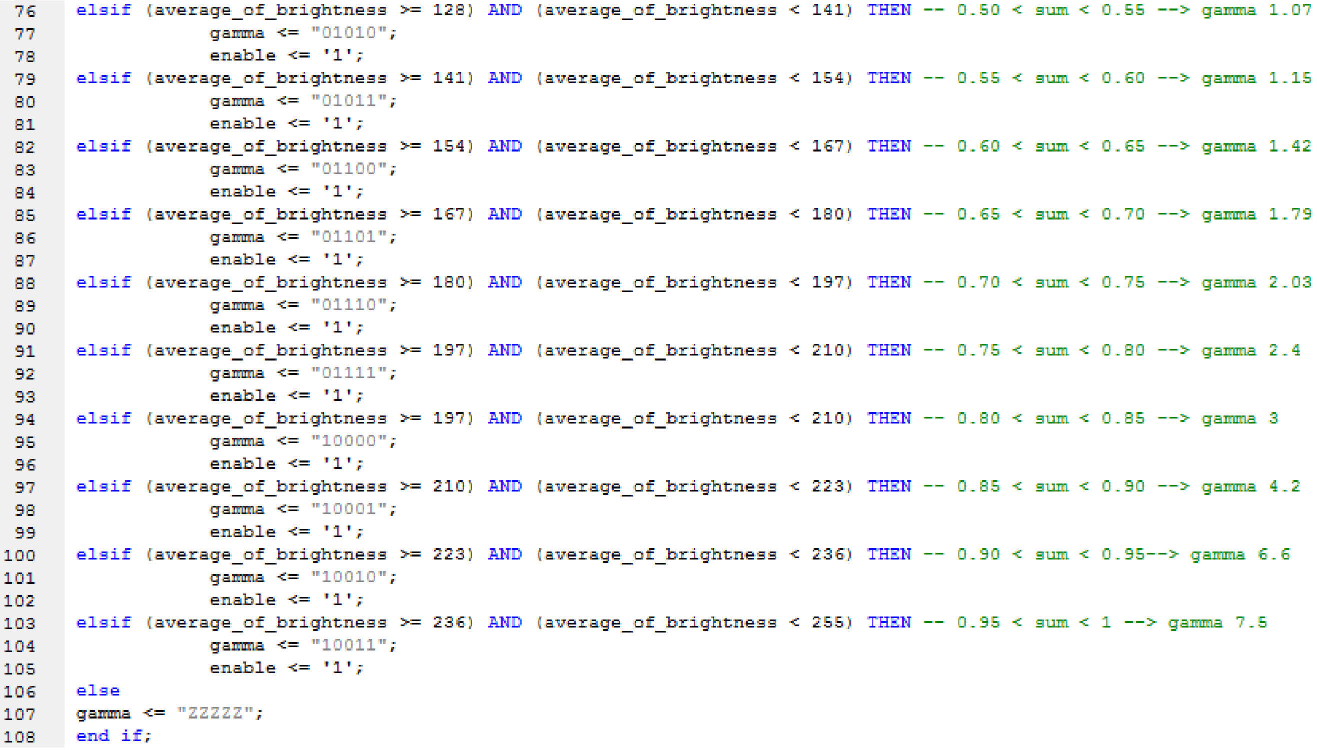


```
entity Gamma_estimator is
port(  input: in std_logic_vector(7 downto 0);
       nrst : in std_logic;
       clk : in std_logic;
       enable : out std_logic;
       gamma: out std_logic_vector(4 downto 0)
       );
end Gamma_estimator;

architecture Behavioral of Gamma_estimator is

signal sum_of_brightness : std_logic_vector(15 downto 0) :="0000000000000000";
signal average_of_brightness : std_logic_vector(7 downto 0);
signal counter : integer:=0;
signal flag : std_logic;
begin
process(clk)
begin
  if(nrst = '1') then
    counter <= 0;
    sum_of_brightness <= "0000000000000000";
    average_of_brightness <="00000000";
    flag <= '0';
  elsif(rising_edge(clk)) then
    sum_of_brightness <= input + sum_of_brightness;
    counter <= counter+ 1;
    if (counter = 256) then
    counter <= 0;
    sum_of_brightness <= "0000000000000000";
    average_of_brightness <= sum_of_brightness(15 downto 8);
    flag <= not flag;
    end if;
 end if;
end process;
```

This part calculates sum of brightness with clk (synchronous).



```vhdl
process(flag)
begin
 if (average_of_brightness >=2) and (average_of_brightness < 13) then -- 0.0 < sum < 0.05 --> gamma 0.2
            gamma <= "00000";
            enable <= '1';
 elsif (average_of_brightness >= 13) AND (average_of_brightness < 26) THEN -- 0.05 < sum < 0.10 --> gamma 0.26
            gamma <= "00001";
            enable <= '1';
 elsif (average_of_brightness >= 26) AND (average_of_brightness < 39) THEN -- 0.10 < sum < 0.15 --> gamma 0.32
            gamma <= "00010";
            enable <= '1';
 elsif (average_of_brightness >= 39) AND (average_of_brightness < 52) THEN -- 0.15 < sum < 0.20 --> gamma 0.37
            gamma <= "00011";
            enable <= '1';
 elsif (average_of_brightness >= 52) AND (average_of_brightness < 65) THEN -- 0.20 < sum < 0.25 --> gamma 0.46
            gamma <= "00100";
            enable <= '1';
 elsif (average_of_brightness >= 65) AND (average_of_brightness < 78) THEN -- 0.25 < sum < 0.30 --> gamma 0.53
            gamma <= "00101";
            enable <= '1';
 elsif (average_of_brightness >= 78) AND (average_of_brightness < 91) THEN -- 0.30 < sum < 0.35 --> gamma 0.62
            gamma <= "00110";
            enable <= '1';
 elsif (average_of_brightness >= 91) AND (average_of_brightness < 104) THEN -- 0.35 < sum < 0.40 --> gamma 0.69
            gamma <= "00111";
            enable <= '1';
 elsif (average_of_brightness >= 104) AND (average_of_brightness < 115) THEN -- 0.40 < sum < 0.45 --> gamma 0.83
             gamma <= "01000";
             enable <= '1';
 elsif (average_of_brightness >= 115) AND (average_of_brightness < 128) THEN -- 0.45 < sum < 0.50 --> gamma 0.96
             gamma <= "01001";
             enable <= '1';
```



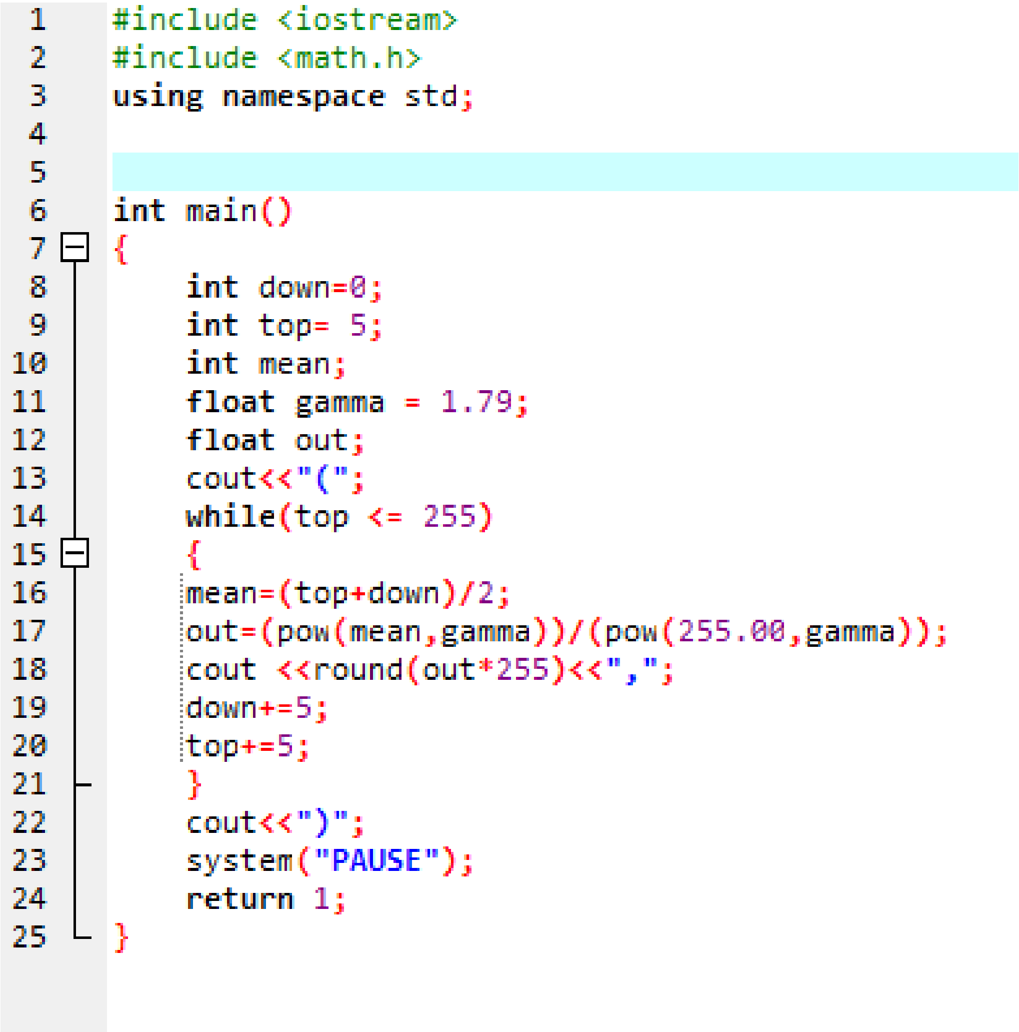
```vhdl
elsif (average_of_brightness >= 128) AND (average_of_brightness < 141) THEN -- 0.50 < sum < 0.55 --> gamma 1.07
            gamma <= "01010";
            enable <= '1';
elsif (average_of_brightness >= 141) AND (average_of_brightness < 154) THEN -- 0.55 < sum < 0.60 --> gamma 1.15
            gamma <= "01011";
            enable <= '1';
elsif (average_of_brightness >= 154) AND (average_of_brightness < 167) THEN -- 0.60 < sum < 0.65 --> gamma 1.42
            gamma <= "01100";
            enable <= '1';
elsif (average_of_brightness >= 167) AND (average_of_brightness < 180) THEN -- 0.65 < sum < 0.70 --> gamma 1.79
            gamma <= "01101";
            enable <= '1';
elsif (average_of_brightness >= 180) AND (average_of_brightness < 197) THEN -- 0.70 < sum < 0.75 --> gamma 2.03
            gamma <= "01110";
            enable <= '1';
elsif (average_of_brightness >= 197) AND (average_of_brightness < 210) THEN -- 0.75 < sum < 0.80 --> gamma 2.4
            gamma <= "01111";
            enable <= '1';
elsif (average_of_brightness >= 197) AND (average_of_brightness < 210) THEN -- 0.80 < sum < 0.85 --> gamma 3
            gamma <= "10000";
            enable <= '1';
elsif (average_of_brightness >= 210) AND (average_of_brightness < 223) THEN -- 0.85 < sum < 0.90 --> gamma 4.2
            gamma <= "10001";
            enable <= '1';
elsif (average_of_brightness >= 223) AND (average_of_brightness < 236) THEN -- 0.90 < sum < 0.95--> gamma 6.6
            gamma <= "10010";
            enable <= '1';
elsif (average_of_brightness >= 236) AND (average_of_brightness < 255) THEN -- 0.95 < sum < 1 --> gamma 7.5
            gamma <= "10011";
            enable <= '1';
else
gamma <= "ZZZZZ";
end if;
```

This part estimate gamma based of average brightness in integer value not float value. This idea reduce area because floating point needs package and handling so needs more area and power in regard to integer value.

5.2. Gamma corrector module

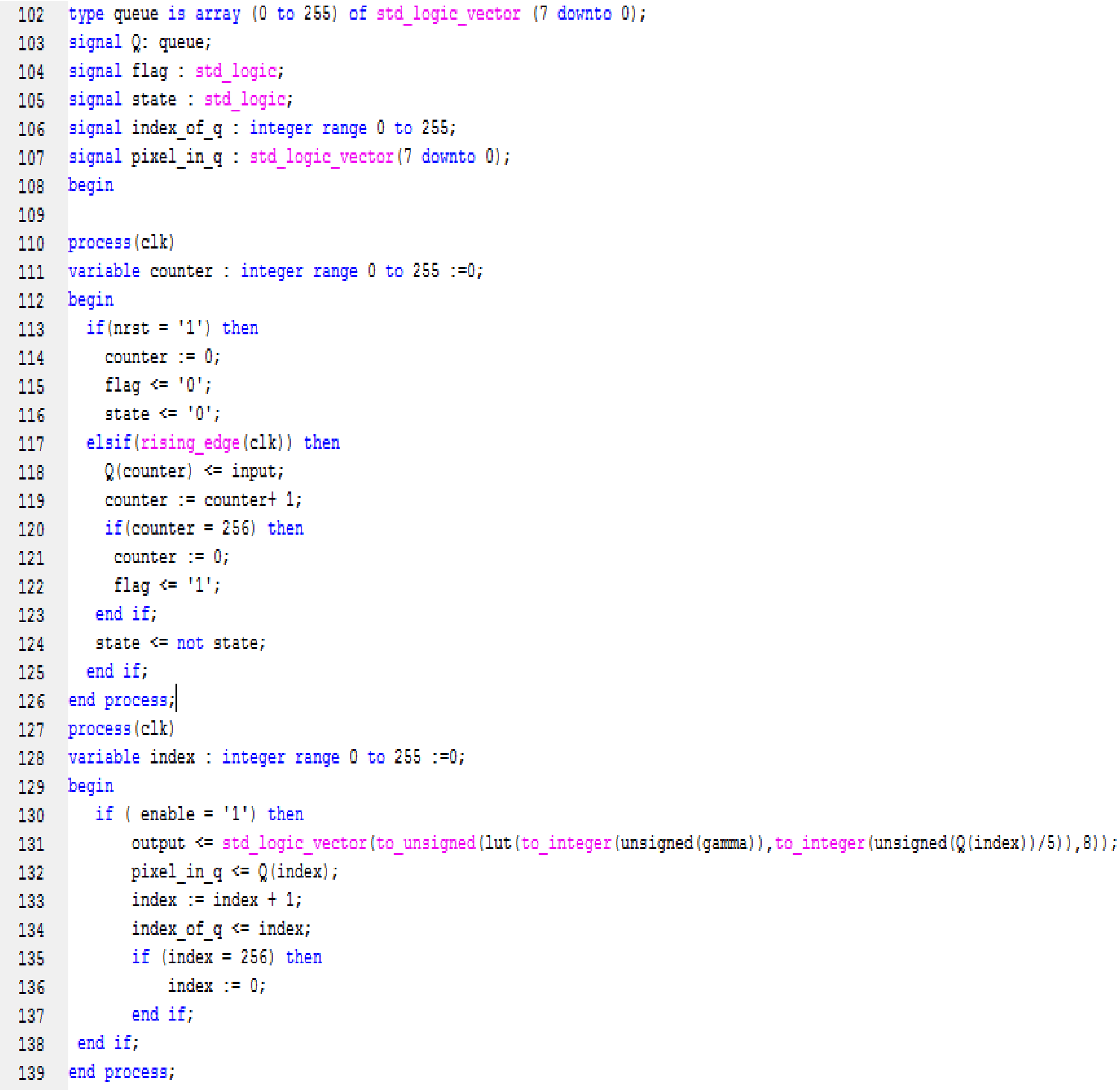
There is a look up table in gamma corrector module. We use look up table because we can't not implement Logarithm and power for numbers between 0 and 1. On the other hand we reduce require time for correction part.

Actually we estimate output based on gamma and input value. We generate look up table with C++ language.



```
#include <iostream>
#include <math.h>
using namespace std;


int main()
{
    int down=0;
    int top= 5;
    int mean;
    float gamma = 1.79;
    float out;
    cout<<"(";
    while(top <= 255)
    {
    mean=(top+down)/2;
    out=(pow(mean,gamma))/(pow(255.00,gamma));
    cout <<round(out*255)<<",";
    down+=5;
    top+=5;
    }
    cout<<")";
    system("PAUSE");
    return 1;
}
```

This code generates look up table based on estimating output.



```
type queue is array (0 to 255) of std_logic_vector (7 downto 0);
signal Q: queue;
signal flag : std_logic;
signal state : std_logic;
signal index_of_q : integer range 0 to 255;
signal pixel_in_q : std_logic_vector(7 downto 0);
begin

process(clk)
variable counter : integer range 0 to 255 :=0;
begin
  if(nrst = '1') then
    counter := 0;
    flag <= '0';
    state <= '0';
  elsif(rising_edge(clk)) then
    Q(counter) <= input;
    counter := counter+ 1;
    if(counter = 256) then
     counter := 0;
     flag <= '1';
   end if;
   state <= not state;
  end if;
end process;
process(clk)
variable index : integer range 0 to 255 :=0;
begin
   if ( enable = '1') then
       output <= std_logic_vector(to_unsigned(lut(to_integer(unsigned(gamma)),to_integer(unsigned(Q(index))/5)),8));
       pixel_in_q <= Q(index);
       index := index + 1;
       index_of_q <= index;
       if (index = 256) then
           index := 0;
       end if;
 end if;
end process;
```

All inputs come to an array (similar to queue), after 256 inputs, gamma is estimated. One clock after gamma ‘event all values in queue will be correct and drive to output. Gamma selects row of look up table and input value selects column of look up table. Column of look up table is estimated based on input value because we divide [0 255] to 51 spans so column value of look up table is input value/51.

7. Reports

Automatic gamma correction based on average brightness in synthesized in cadence and encounter.

7.1 Timing report

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : automatic_gamma_correction
Version: C-2009.06-SP5
Date   : Mo Feb 8 22:33:03 2016
****************************************

Operating Conditions: typical   Library: typical
Wire Load Model Mode: top

  Startpoint: gamma_estimator_module/counter_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: gamma_estimator_module/counter_reg[31]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  automatic_gamma_correction
                     tsmc18_wl40           typical

  Point                                                   Incr
Path
  --------------------------------------------------------------------
------
  clock clk (rise edge)                                  10.00
30.00
  clock network delay (ideal)                             0.20
30.20
  gamma_estimator_module/counter_reg[0]/CK (DFFRHQX1)     0.00
30.20 r
  gamma_estimator_module/counter_reg[0]/Q (DFFRHQX1)      1.12
31.32 r
  gamma_estimator_module/add_36/A[0] (Gamma_estimator_DW01_inc_0)
                                                          0.00
31.32 r
  gamma_estimator_module/add_36/U1_1_1/CO (ADDHXL)        0.35
31.67 r
  gamma_estimator_module/add_36/U1_1_2/CO (ADDHXL)        0.39
32.06 r
```

```
  gamma_estimator_module/add_36/U1_1_3/CO (ADDHXL)        0.39
32.45 r
  gamma_estimator_module/add_36/U1_1_4/CO (ADDHXL)        0.39
32.83 r
  gamma_estimator_module/add_36/U1_1_5/CO (ADDHXL)        0.39
33.22 r
  gamma_estimator_module/add_36/U1_1_6/CO (ADDHXL)        0.39
33.60 r
  gamma_estimator_module/add_36/U1_1_7/CO (ADDHXL)        0.39
33.99 r
  gamma_estimator_module/add_36/U1_1_8/CO (ADDHXL)        0.39
34.38 r
  gamma_estimator_module/add_36/U1_1_9/CO (ADDHXL)        0.39
34.76 r
  gamma_estimator_module/add_36/U1_1_10/CO (ADDHXL)       0.39
35.15 r
  gamma_estimator_module/add_36/U1_1_11/CO (ADDHXL)       0.39
35.53 r
  gamma_estimator_module/add_36/U1_1_12/CO (ADDHXL)       0.39
35.92 r
  gamma_estimator_module/add_36/U1_1_13/CO (ADDHXL)       0.39
36.31 r
  gamma_estimator_module/add_36/U1_1_14/CO (ADDHXL)       0.39
36.69 r
  gamma_estimator_module/add_36/U1_1_15/CO (ADDHXL)       0.39
37.08 r
  gamma_estimator_module/add_36/U1_1_16/CO (ADDHXL)       0.39
37.46 r
  gamma_estimator_module/add_36/U1_1_17/CO (ADDHXL)       0.39
37.85 r
  gamma_estimator_module/add_36/U1_1_18/CO (ADDHXL)       0.39
38.24 r
  gamma_estimator_module/add_36/U1_1_19/CO (ADDHXL)       0.39
38.62 r
  gamma_estimator_module/add_36/U1_1_20/CO (ADDHXL)       0.39
39.01 r
  gamma_estimator_module/add_36/U1_1_21/CO (ADDHXL)       0.39
39.39 r
  gamma_estimator_module/add_36/U1_1_22/CO (ADDHXL)       0.39
39.78 r
  gamma_estimator_module/add_36/U1_1_23/CO (ADDHXL)       0.39
40.17 r
  gamma_estimator_module/add_36/U1_1_24/CO (ADDHXL)       0.39
40.55 r
  gamma_estimator_module/add_36/U1_1_25/CO (ADDHXL)       0.39
40.94 r
  gamma_estimator_module/add_36/U1_1_26/CO (ADDHXL)       0.39
41.32 r
  gamma_estimator_module/add_36/U1_1_27/CO (ADDHXL)       0.39
41.71 r
```

```
  gamma_estimator_module/add_36/U1_1_28/CO (ADDHXL)       0.39
42.10 r
  gamma_estimator_module/add_36/U1_1_29/CO (ADDHXL)       0.39
42.48 r
  gamma_estimator_module/add_36/U1_1_30/CO (ADDHXL)       0.38
42.87 r
  gamma_estimator_module/add_36/U2/Y (XOR2X1)             0.45
43.32 r
  gamma_estimator_module/add_36/SUM[31] (Gamma_estimator_DW01_inc_0)
                                                          0.00
43.32 r
  gamma_estimator_module/U165/Y (AND2X1)                  0.32
43.63 r
  gamma_estimator_module/counter_reg[31]/D (DFFRHQXL)     0.00
43.63 r
  data arrival time
43.63

  clock clk (rise edge)                                  30.00
30.00
  clock network delay (ideal)                             0.20
30.20
  clock uncertainty                                       0.10
30.10
  gamma_estimator_module/counter_reg[31]/CK (DFFRHQXL)
                                                          0.00
30.10 r
  library setup time                                      0.08
30.02
  data required time
30.02
  --------------------------------------------------------------------
------
  data required time
30.02
  data arrival time
43.63
  --------------------------------------------------------------------
------
  slack (MET)
70.65
```

7.2 Power report

```
Loading db file
'/home/icic/Desktop/automatic_gamma_correction/typical.db'
Information: Propagating switching activity (low effort zero delay
simulation). (PWR-6)
Warning: Design has unannotated primary inputs. (PWR-414)
Warning: Design has unannotated sequential cell outputs. (PWR-415)
```

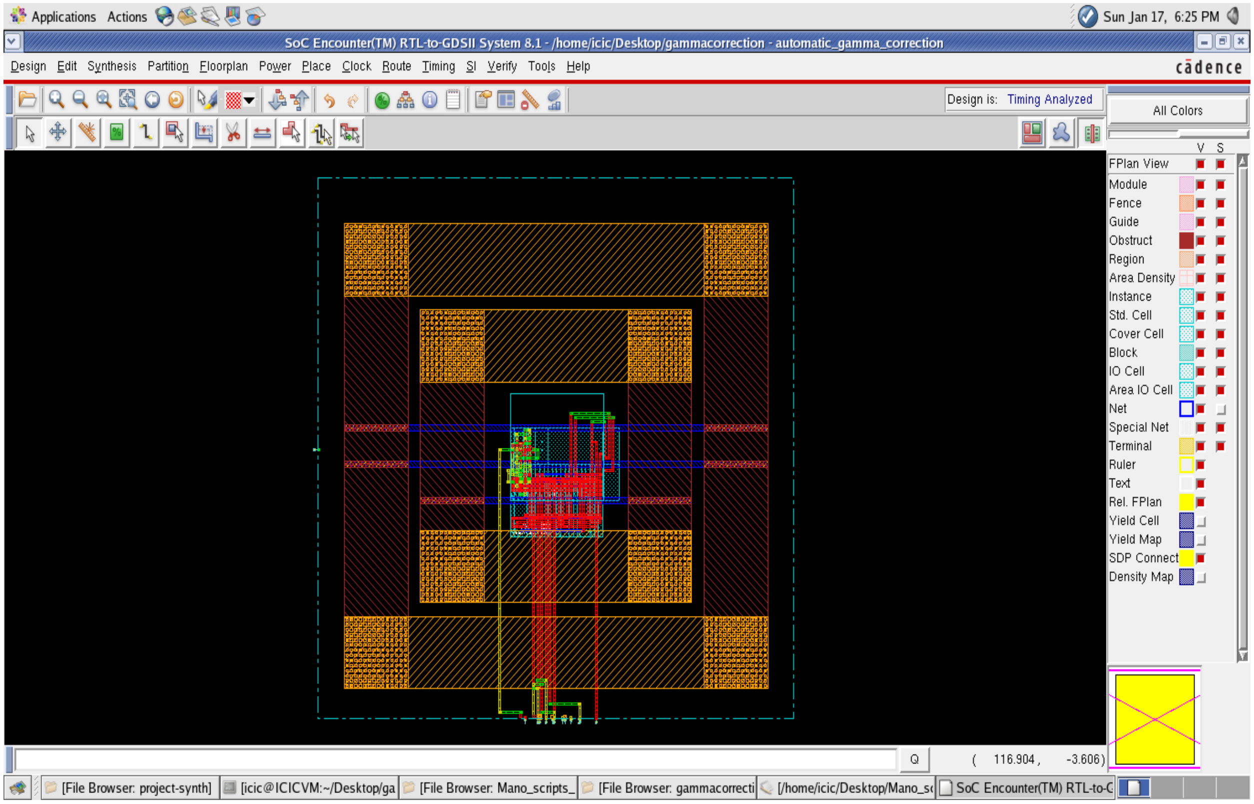
```
****************************************
Report : power
        -analysis_effort low
Design : automatic_gamma_correction
Version: C-2009.06-SP5
Date   : Sun Jan 17 22:33:04 2016
****************************************


Library(s) Used:

    typical (File:
/home/icic/Desktop/automatic_gamma_correction/typical.db)


Operating Conditions: typical   Library: typical
Wire Load Model Mode: top

Design        Wire Load Model            Library
------------------------------------------------
automatic_gamma_correction
                       tsmc18_wl40       typical


Global Operating Voltage = 1.8
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  = 237.1949 uW   (73%)
  Net Switching Power  =  89.4683 uW   (27%)
                         ---------
Total Dynamic Power    = 326.6631 uW  (100%)

Cell Leakage Power     =  43.0993 nW
```

7.3 Area report

```
****************************************
Report : area
Design : automatic_gamma_correction
Version: C-2009.06-SP5
Date   : Sun Jan 17 22:33:03 2016
****************************************
```

```
Library(s) Used:

    typical (File:
/home/icic/Desktop/automatic_gamma_correction/typical.db)

Number of ports:              18
Number of nets:               24
Number of cells:               2
Number of references:          2

Combinational area:       4141.368064
Noncombinational area:    4111.430414
Net Interconnect area:    124507.287720

Total cell area:          8252.798478
Total area:               132760.086198
```

7.3 Layout


Applications Actions
Sun Jan 17, 6:25 PM
SoC Encounter(TM) RTL-to-GDSII System 8.1 - /home/icic/Desktop/gammacorrection - automatic_gamma_correction
Design Edit Synthesis Partition Floorplan Power Place Clock Route Timing SI Verify Tools Help
cādence
Design is: Timing Analyzed
All Colors
V S
FPlan View
Module
Fence
Guide
Obstruct
Region
Area Density
Instance
Std. Cell
Cover Cell
Block
IO Cell
Area IO Cell
Net
Special Net
Terminal
Ruler
Text
Rel. FPlan
Yield Cell
Yield Map
SDP Connect
Density Map
Q
(116.904 , -3.606)
[File Browser: project-synth]
[icic@ICICVM:~/Desktop/ga
[File Browser: Mano_scripts_
[File Browser: gammacorrecti
[/home/icic/Desktop/Mano_s
SoC Encounter(TM) RTL-to-G

6. Conclusion

We have introduced a new image enhancement method based on gamma correction that estimates image gamma values without any calibration information or knowledge of the imaging device. The proposed method is a necessary preprocessing stage for most image analysis. Experimental results in this research indicate that the proposed method improves image quality, enhances the dynamic range and details of the image in least amount of time. On the other hand, proposed method performs in less time than other methods. This method can be implemented as a ASIC in the photography or printing devices. On the other hand, we reduce Latency to 1/256 of past latency with dividing image to smaller blocks and estimate gamma based on proposed method and enhanced them.

References

[1] R.P. Kleihorst, R.L. Lagendiik, and J. Biemond. An adaptive order-statistic noise _lter for gammacorrected image sequences. *IEEE Transactions on Image Processing*, 6(10):1442.1446, 1997.

[2] Pizurica, A. and Philips, W. “Estimating the probability of the presence of a signal of interest in multiresolution single and multiband image denoising”, ***IEEE Trans. Image Process***, 654–665, 2006.

[3] Shi, Y., Yang, J. and Wu, R. "Reducing Illumination Based on Nonlinear Gamma Correction," ***In Proc.*** **310** - Vol. 24, No. 4, 2011.

[4] Gonzalez, R. C. and Woods, R. E. Digital Image Processing. Prentice Hall, Upper Saddle River, NJ 07458, 2002.

[5] Farid, H. "Blind inverse gamma correction", ***IEEE Transactions on Image Processing***, Vol. 10, pp. 1428-1433, 2001.

[6] Pizer, S. M., E.P. Ambum and J.D. Austin, “Adaptive histogram equalization and its variation”, ***Computer Vision, Graphics, and Image Processing*** Vol. 39, No. 3, 355–368, 1987.

[7] Chen, S. D. and Ramli, A. R. “Minimum mean brightness error bihistogram equalization in contrast enhancement”, ***IEEE Transactions on Consumer Electronics,*** Vol. 49, No. 4, 1310–1319, 2003.

[8] FarshbafDoustar, M. and Hassanpour, H. "A Locally- Adaptive Approach For Image Gamma Correction," 10th International Conference on Information Sciences, ***Signal Processing and their Applications (ISSPA2010)*** 73-76. (2010)

[9] Asadi Amiri, S., Hassanpour ,H. and Pouyan, A.“Texture Based Image Enhancement Using Gamma Correction”, ***Middle-East Journal of Scientific Research*** , Vol. 6, 569-574. (2010)

[10] Guillon, S., Baylou, P., Najim, M. and Keskes, N. “Adaptive nonlinear filters for 2D a nd 3D image enhancement”, ***Signal Processing***, Vol. 67, No. 3 ,237–254, 1998.

[11] Babakhani, P. “automatic gamma correction based on average and deviation from center of histogram”. 2th International Conference on Advances in Engineering and Basic Sciences,76 – 82, 2015.

[12] Hasanpour, H. Asadi Amiri, S. Pouyan, A, A. “Automatic brighness enhancement based on local gamma correction”. National conference of computer engineering and information technology. Hamedan, 2011.