Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-3-2006

Color Calibration for Multi-Camera System by using Color Pattern Board

K. Yamamoto and J. U

Color Calibration for Multi-Camera System by using Color Pattern Board

K. Yamamoto and J. U

17th February, 2006

Abstract

Color calibration is necessary for multi-camera system. In this document, we introduce our color calibration approach that was developed in Monash university when Kenji Yamamoto was an exchange student. In this method we put a color pattern board in front of multi-camera system before/after capturing target videos. After capturing this board by all cameras at once, we create a look-up table that indicates the modification from captured video to color calibrated video. We modify the target video by using this look-up table as post-processing. Experimental results show that this approach works well especially for the colors that are similar with the colors in color pattern board.

1 Introduction

As the ability of personal computers and manufacturing cameras continues to dramatically advance, multi-camera systems have emerged all over the world[1, 2]. Nowadays they have attracted a great deal of attention, finding multiple fields in which to be applied: the preservation of cultural heritage and traditional dancing, free viewpoint television, 3-D world, educational applications, entertainment *et al.* To realise these applications, camera calibration, color calibration, efficient video coding *et al* are necessary.

Some color calibration have been developed for multi-camera systems. In [3] the captured images are modified by using the histogram at matching areas between two cameras. These histograms are created for each camera that should be calibrated. In [4] in order to use H.264/AVC for multi-camera coding, average and variance are used in block matching process. In [5] Kawai assumed linear modification model, and modify captured image iterately by using some coefficients that are calculated by segmentation.

On this document we introduce a color calibration method that uses a color pattern board. In general, accuracy of camera calibration becomes better if we use a camera calibration board. It is the same in the case of color calibration. The first reason is that we can calculate correspondences between cameras without error. The second reason is that we can choose the colors on the board to our own design to cover whole color space.

The three color channels of a video such as R, G and B have a little relation between them in general, if the video is captured by 1CCD camera. However in this method we ignore this relations and consider how to calibrate three channels independently.

Because the speed of the data coming from CCD device is very high, the A/D converter that converts the CCD analog data to digital output can not convert them independently. This means the last converted datum affects the next datum. In the case of 1CCD camera that uses general Bayer-pattern, color channels affect each other. However in the case of multi-camera system here, the CCD analog data are transformed to RGB channels automatically in consideration for CCD device. Therefore we ignore this problem and treat RGB channels independently.

In this method we decide a reference camera at first [1]. The captured videos of other cameras(target cameras) will be modified to be similar with it. The calculation steps are as follows:

1. **abstraction of the colors from captured color pattern board in all camera**

   The software 'pickup_colors.exe' executes this step for one camera at a time.

[1]It is possible not to decide a reference camera. In this case virtual data which are used as reference camera is necessary.

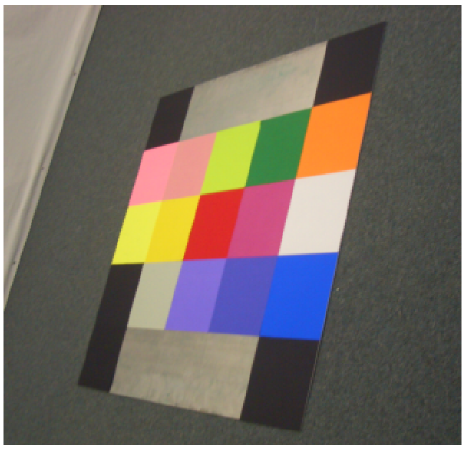


Figure 1: Color Pattern Board

2. **calculation of the look-up tables that indicates the modification from captured videos to color calibrated videos**

   We caluculate one look-up table for one target camera. In this document we introduce two approaches. The software 'rgb_independence.exe' and 'DP_color_calibration.exe' execute this step and the following step for one camera at a time.

3. **modification of the videos**

We will describe these steps exactly from section 2 to section 4. Section 5 shows experimental result. We present discussion and conclusion in section 6 and 7 respectively.

2 Abstraction of the Colors

We put a color pattern board, which consists of dozens of color patches, in front of multi-camera system. Before/after capturing videos, we capture it by all cameras at once. Assume that the total of cameras is C, and the total of color patches is N. The captured color patches are presented by

$$\begin{gathered} I_R(c,n), \\ I_G(c,n), \\ I_B(c,n), \end{gathered} \tag{1}$$

where I_R, I_G and I_B represent intensities of R, G and B channel respectively. $c \in (0, 1, ..., C-1)$ represents the camera index and $n \in (0, 1, ..., N-1)$ represents the color patch index. To calculate the best look-up tables in the section 3, it is recommended to use the average of intensities inside patches as I_R *et al.* It is also recommended to scatter the colors in order to get variety of intensities in the R, G and B channel.

Table 1: Example of the Relations between Color Patch Index and Intensity

"camera #0"

color patch index	$I_R(0, n)$
0	51
1	199
2	102
...	...
N-1	98

color patch index	$I_G(0, n)$
0	0
1	22
2	241
...	...
N-1	10

color patch index	$I_B(0, n)$
0	201
1	1
2	6
...	...
N-1	3

"camera #1"

color patch index	$I_R(1, n)$
0	63
1	195
2	99
...	...
N-1	90

color patch index	$I_G(1, n)$
0	0
1	24
2	222
...	...
N-1	15

color patch index	$I_B(1, n)$
0	255
1	4
2	12
...	...
N-1	8

"camera #2"
...
"camera #(C-1)"
...

The goal of this step is to make the relation between color patch indices and I_R *et al.* Example of the color pattern board and the goal of this step is shown in Figure 1 and Table (1) respectively.

The software 'pickup_colors.exe' executes this step. How to use this software is depicted in Appendix A.

3 Calculation of the Look-up Tables

By using the results of the section 2, we create the look-up tables that indicates the modification from captured videos to color calibrated videos. Three look-up tables are created between each target camera and reference camera,

$$\begin{aligned} v'_R &= T_R(c_{tgt}, v_R), \\ v'_G &= T_G(c_{tgt}, v_G), \\ v'_B &= T_B(c_{tgt}, v_B), \end{aligned} \tag{2}$$

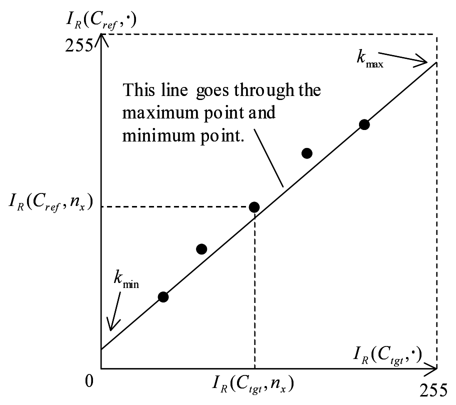


Figure 2: k_{min} and k_{max}

where T_R, T_G and T_B represent the look-up tables of R, G and B channel respectively. $c_{tgt} \in (0, 1, ..., C-1)$ stands for the target camera index. v_R, v_G, v_B denote the intensity of captured videos, and v'_R, v'_G, v'_B denote the intensity of color calibrated videos.

3.1 First Approach: Linear Interpolation

To create T_R, T_G and T_B, we assume the following conditions:

- The corresponding patches should be same. This is formulated as follows:

$$\begin{aligned} I_R(c_{ref}, n) = I_R(c_{tgt}, n), \\ I_G(c_{ref}, n) = I_G(c_{tgt}, n), \\ I_B(c_{ref}, n) = I_B(c_{tgt}, n), \end{aligned} \tag{3}$$

  where $c_{ref}, c_{tgt} \in (0, 1, ..., C-1), c_{ref} \neq c_{tgt}$ and $n \in (0, 1, ..., N-1)$.

- If there is not the relation about 0 intensity in R channel of the target camera, the following relation exists in addition.

$$\begin{aligned} I_R(c_{tgt}, maximum(n)+1) &= 0, \\ I_R(c_{ref}, maximum(n)+1) &= k_{min}, \end{aligned} \tag{4}$$

  where k_{min} shows in Figure 2. The relation around the k_{min} will be sensitive, because k_{min} is not a captured datum. To void this sensitivity as far as possible, we should capture small intensity. This addition is done in the same way in G and B channel respectively.

- If there is not the relation about 255 intensity in R channel of the target camera, the following relation exists in addition.

$$I_R(c_{tgt}, maximum(n)+1) = 255, \tag{5}$$

$$I_R(c_{ref}, maximum(n)+1) \quad = \quad k_{max},$$

where k_{max} shows in Figure 2. The relation around the k_{max} will be sensitive, because k_{max} is not a captured datum. To void this sensitivity as far as possible, we should capture big intensity. This addition is done in the same way in G and B channel respectively.

Actually we calculate as following order to create T_R *et al.* Example of this calculation is shown in Table (2)-(5). This example assumes that camera #0 is the reference camera and camera #1 is the target camera.:

1. Create the temporal table of R channel that consists of the intensities of the target camera and those of reference camera. Temporal tables of G and B channel are created in the same way respectively. See Table (2).
2. Insert the relation mentioned in equation (4) if there is not the relation about 0 intensity in R channel of the target camera. This insertion is done in the same way in G and B channel respectively. See Table (3).
3. Insert the relation mentioned in equation (5) if there is not the relation about 255 intensity in R channel of the target camera. This insertion is done in the same way in G and B channel respectively. See Table (3).
4. Sort the relation in R channel by the intensity of the target camera. This sorting is done in the same way in G and B channel respectively. See Table (4).
5. Interpolate the intensity in R channel by linear interpolation and create T_R. This Interpolation is done to create T_G and T_B in the same way in G and B channel respectively. See Table (5).

The software 'rgb_independence.exe' executes this calculation. How to use this software is shown in Appendix B.

3.2 Second Approach: Energy Function

To create T_R, T_G and T_B, we assume an energy function E and calculate minimum by using dynamic programming. The energy function consists of two terms, E_α and E_β. E_α means that the equation (4) should be true. If not, their differences increase the energy. E_β means that the intensities should increase step by step. If not, the differences increase the energy. E is formulated as follows:

$$\begin{aligned} T_R(c_{tgt}, v) &= \arg\min_{T_R} E(c_{tgt}, v), \\ E(c_{tgt}, v) &= \alpha \cdot E_\alpha(c_{tgt}, v) + \beta \cdot E_\beta(c_{tgt}, v), \\ E_\alpha(c_{tgt}, v) &= sum_{n=0}^{N-1} |I_R(c_{ref}, n) - T_R(c_{tgt}, I_R(c_{tgt}, n))|, \qquad (6) \\ E_\beta(c_{tgt}, v) &= sum_{m=1}^{255} |T_R(c_{tgt}, m) - T_R(c_{tgt}, m-1) - 1|, \end{aligned}$$

where α and β are constant parameters. N is the total of correspondences. T_G and T_B are not mentioned here but we define them in the same way.

The software 'DP_color_calibration.exe' executes this calculation. How to use this software is shown in Appendix C.

4 Modification of the Videos

By using the results of the section 3, we modify captured videos to calibrated videos. The modification of each pixel are shown in equation (2). We apply this equation to all pixels in the videos as follows:

$$\begin{aligned} v'_R(c_{tgt}, w, h, t) &= T_R(c_{tgt}, v_R(c_{tgt}, w, h, t)), \\ v'_G(c_{tgt}, w, h, t) &= T_G(c_{tgt}, v_G(c_{tgt}, w, h, t)), \\ v'_B(c_{tgt}, w, h, t) &= T_B(c_{tgt}, v_B(c_{tgt}, w, h, t)), \end{aligned} \tag{7}$$

where $w(0 \leq w < width)$,$h(0 \leq h < height)$ and $t(0 \leq t < time)$ stands for the pixel position and time respectively.

The software 'rgb_independence.exe' and 'DP_color_calibration.exe' executes this step. How to use this software is shown in Appendix B.

5 Experimental Result

To confirm this method, we conducted an experiment. Because setting up for video capturing spends much time, we conduct an experiment only the still images. The condition of this experiment are as follows:

- use 4 cameras ($C = 4$).
- use the color pattern board shown in figure 1. This board have 25 patches inside. The 10 patches located at upper row and lower row are used just for detecting patch positions. Other 15 patches are used for color calibration ($N = 15$).
- capture color pattern board by 4 cameras at once.
- capture human beings by 4 cameras at once.
- use camera #0 as a reference camera. Because the image captured camera #0 seems better than others, we use this camera as a reference camera.

Figure 3 and figure 4 represent the captured color pattern board and human beings respectively. Figure 5 and figure 6 depict the color calibrated ones by the first approach(linear interpolation). Figure 7 and figure 8 depict the color calibrated ones by the second approach(energy function). As you see, this color calibration works well especially for the colors that are similar with the colors in color pattern board.

In this results the first approach seems better than the second one. It is the reason that the equation (5) works well for bright background.

6 Discussion

As you see, the background color in Figure 5 and figure 6 are different. The reason is that there are no color information in color pattern board for the background color. Figure 9 present the example of this problem. In this figure there are two areas. One is the narrow color area in color pattern board, and the other is the wide captured whole color area. In this case, this color calibration does not work well in wide area. To avoid this problem, to use vivid color pattern board and to capture it at bright area are necessary.

Comparing the first approach and the second approach, we feel as though the first approach is better. But it is just true in this example pictures. To predict the color in wide area works well by good fortune.

7 Conclusion

In this document, we introduce our color calibration method that use a color calibration pattern. By using it we can escape the correspondence problem. The experiment conducted shows good results especially for the colors that are similar with the colors in color pattern board. In order to evaluate the accuracy and reliability of color calibration, we have to establish how to measure the quality of color calibration.

References

[1] Bennett Wilburn, Michael Smulski, Kelin Lee, and Mark A. Horowitz: "The Light Field Video Camera," SPIE Electronic Imaging Conference on Media Processors 2002 (2002)

[2] Masayuki Tanimoto, Toshiaki Fujii, and Terumasa Aoki: "Test Sequences for Call for Proposals on Multiview Video Coding," ISO/IEC JTC1/SC29/WG11 M12022 (2005)

[3] Fatih Porikli: "Inter-Camera Color Calibration by Correlation Model Function," Mitsubishi Electric Research Laboratory Technical Report, TR-2003-103 (2004)

[4] Joaquin L'opez, Jae Hoon Kim, Antonio Ortega, and George Chen: "Block-based Illumination Compensation and Search Techniques for Multiview Video Coding", Picture Coding Symposium 2004 (2004)

[5] Yoshihiro Kawai and Fumiaki Tomita: "Intensity Calibration for Stereo Images Based on Segment Correspondence", IAPR Workshop on Machine Vision Applications, pp.331–334, Chiba, Japan (1998)

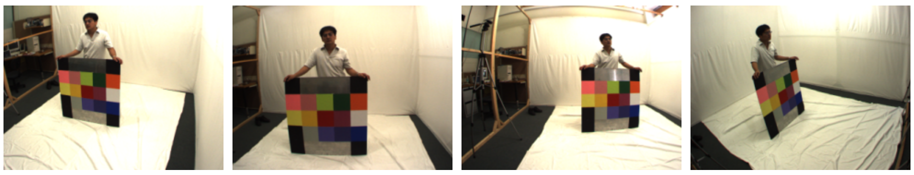

(a) camera #0 (b) camera #1 (c) camera #2 (d) camera #3

Figure 3: Captured Color Pattern Board

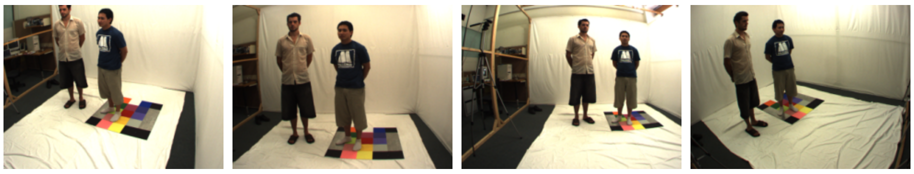

(a) camera #0 (b) camera #1 (c) camera #2 (d) camera #3

Figure 4: Captured Human Beings

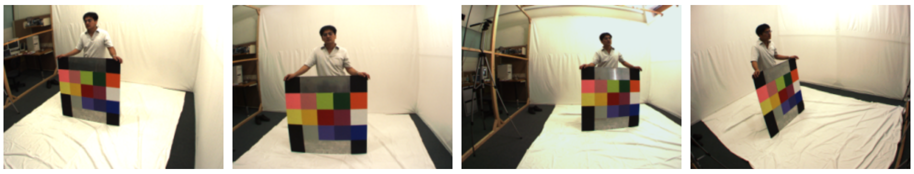
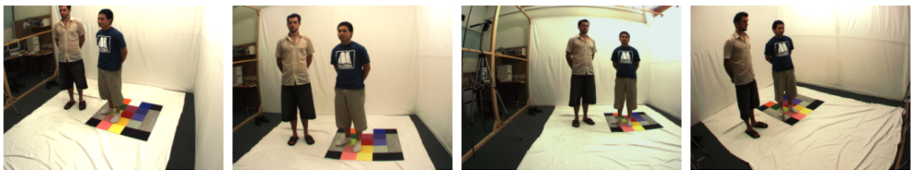
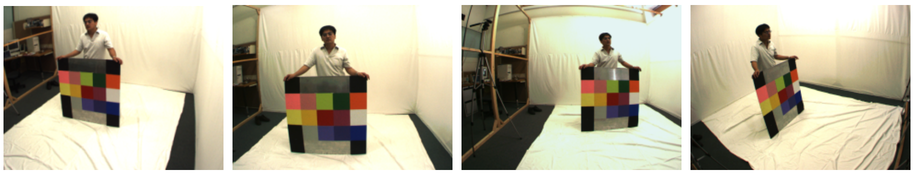

(a) camera #0 equals to the captured image. (b) camera #1 (c) camera #2 (d) camera #3

Figure 5: Color Pattern Board Calibrated by First Approach (Linear Interpolation)



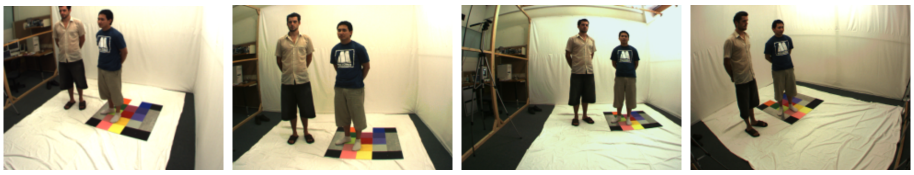
(a) camera #0 equals to the captured image. (b) camera #1 (c) camera #2 (d) camera #3

Figure 6: Human Beings Calibrated by First Approach (Linear Interpolation)



(a) camera #0 equals to the captured image. (b) camera #1 (c) camera #2 (d) camera #3

Figure 7: Color Pattern Board Calibrated by Second Approach (Energy Function)



(a) camera #0 equals to the captured image. (b) camera #1 (c) camera #2 (d) camera #3

Figure 8: Human Beings Calibrated by Second Approach (Energy Function)

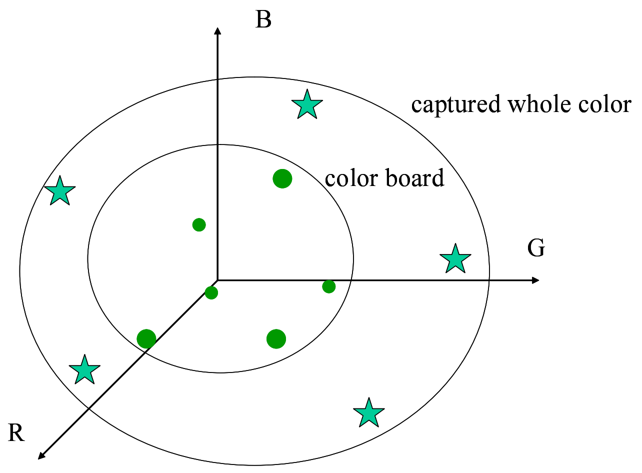
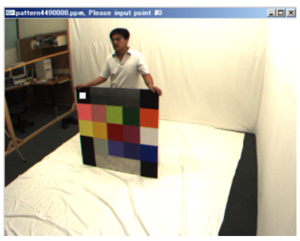
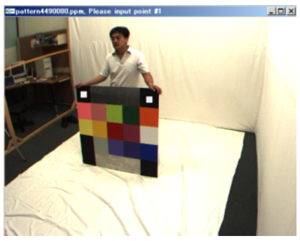


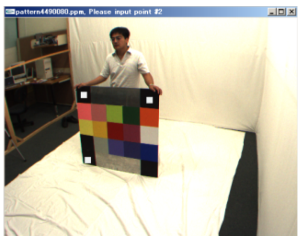
Figure 9: color space


pattern4490080.ppm, Please input point #0

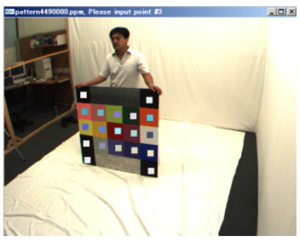
(a) first point: upper_left


pattern4490080.ppm, Please input point #1

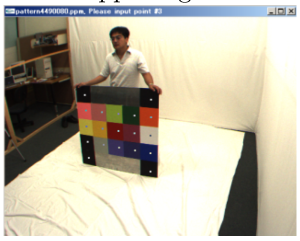
(b) second point: upper_right


pattern4490080.ppm, Please input point #2

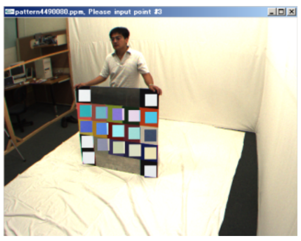
(c) third point: down_left


pattern4490080.ppm, Please input point #3

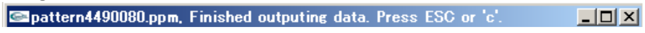
(d) fourth point: down_right


pattern4490080.ppm, Please input point #3

(e) small area


pattern4490080.ppm, Please input point #3

(f) large area


pattern4490080.ppm, Finished outputting data. Press ESC or 'c'.

(g) output indicator

Figure 10: pickup_color.exe

Appendix

A pickup_colors.exe

This program uses OpenGL and GLUT and runs on WindowsXP. Please type the following in a CMD window.

pickup_colors.exe color_ptn.ppm (return)

where color_ptn.ppm is the captured color pattern board image.

Then the image window will appear. On this window you must set the four control points.

The first point is the upper left point. Please type '0' on keyboard and click the center of upper left black patch. The clicked point will be changed such as figure 10 (a).

The second point is the upper right point. Please type '1' on keyboard and click the center of upper right black patch. The clicked point will be changed such as figure 10 (b).

The third point is the down left point. Please type '2' on keyboard and click the center of down left black patch. The clicked point will be changed such as figure 10 (c).

The fourth point is the down right point. Please type '3' on keyboard and click the center of down right black patch. The clicked point will be changed such as figure 10 (d).

When you want to turn down the block area, type 's' on keyboard. The area will turn down like figure 10 (e). When you want to enlarge the block area, type 'l' on keyboard. The area will enlarge like figure 10 (f). It is recommended to enlarge the block as much as you can.

When you finish choosing area, type 'o' on keyboard. The file will be output and the window title will change like figure 10 (g).

If you want to change again, please type '0', '1', '2', '3', 's' or 'l'. You can modify whenever you want.

When you want to quit, type ESC key.

B rgb_independence.exe

This program runs on WindowsXP. Please type the following in a CMD window.

```
rgb_independence.exe tgt.txt ref.txt tgt.ppm out.ppm (return)
```

where tgt.txt is the color pattern board data of a target camera, which is output of **pickup_colors.exe**. ref.txt is the color pattern board data of a reference camera, which is output of **pickup_colors.exe**. tgt.ppm is a captured image of a target camera that will be calibrated. out.ppm is the output of this program.

When you want to calibrate video, please repeat this software.

C DP_color_calibration.exe

This program runs on WindowsXP. Please type the following in a CMD window.

```
DP_color_calibration.exe target.cfg (return)
```

where target.cfg is the configuration file.

When you want to calibrate video, please repeat this software. Example of this configureation file is as follows. The charactors after '#' are treated as comments.

```
#
# DP_color_calibration.cfg
#
#### create look-up table ####
base_colors = "pattern4490080.txt"
target_colors = "pattern4490101.txt"
target_yuv = "4490101.yuv"
height = 768
width = 1024
frame = 0
alpha = 1 # This should be 1.
beta = 0.01
#### output ####
output_fname = "r_4490101"
out_corres = 1 # 0: no output 1: output, corres: correspondence
out_yuv = 0 # 0: no output 1: output
out_ppm = 1 # 0: no output 1: output
out_r_map = 0 # 0: no output 1: output
out_g_map = 0 # 0: no output 1: output
out_b_map = 0 # 0: no output 1: output
out_r_table = 0 # 0: no output 1: output
out_g_table = 0 # 0: no output 1: output
out_b_table = 0 # 0: no output 1: output
```

Table 2: Example of temporal $T_R(1)$,$T_G(1)$ and $T_B(1)$

temporal $T_R(1)$

color patch index	target cam. #1	ref. cam. #0
0	63	51
1	195	199
2	99	102
...	...	...
N-1	90	98

temporal $T_G(1)$

color patch index	target cam. #1	ref. cam. #0
0	0	0
1	24	22
2	222	241
...	...	...
N-1	15	10

temporal $T_B(1)$

color patch index	target cam. #1	ref. cam. #0
0	255	201
1	4	1
2	12	6
...	...	...
N-1	8	3

Table 3: Example of temporal $T_R(1)$,$T_G(1)$ and $T_B(1)$ after insertion

temporal $T_R(1)$

color patch index	target cam. #1	ref. cam. #0
0	63	51
1	195	199
2	99	102
...	...	...
N-1	90	98
-	0	0
-	255	255

temporal $T_G(1)$

color patch index	target cam. #1	ref. cam. #0
0	0	0
1	24	22
2	222	241
...	...	...
N-1	15	10
-	255	255

temporal $T_B(1)$

color patch index	target cam. #1	ref. cam. #0
0	255	201
1	4	1
2	12	6
...	...	...
N-1	8	3
-	0	0

Table 4: Example of temporal $T_R(1)$,$T_G(1)$ and $T_B(1)$ after sorting

temporal $T_R(1)$

color patch index	target cam. #1	ref. cam. #0
-	0	0
...	...	...
0	63	51
...	...	...
N-1	90	98
...	...	...
2	99	102
...	...	...
1	195	199
...	...	...
-	255	255

temporal $T_G(1)$

color patch index	target cam. #1	ref. cam. #0
0	0	0
...	...	...
N-1	15	10
...	...	...
1	24	22
...	...	...
2	222	241
...	...	...
-	255	255

temporal $T_B(1)$

color patch index	target cam. #1	ref. cam. #0
-	0	0
...	...	...
1	4	1
...	...	...
N-1	8	3
...	...	...
2	12	6
...	...	...
0	255	201

Table 5: Example of $T_R(1)$,$T_G(1)$ and $T_B(1)$

temporal $T_R(1)$

color patch index	target cam. #1	ref. cam. #0
-	0	0
-	1	1
-	2	2
-	3	3
-	4	4
-	5	4
-	6	5
...	...	...
0	63	51
...	...	...
N-1	90	98
...	...	...
2	99	102
...	...	...
1	195	199
...	...	...
-	255	255

temporal $T_G(1)$

color patch index	target cam. #1	ref. cam. #0
0	0	0
-	1	1
-	2	2
-	3	2
-	4	3
-	5	4
-	6	4
...	...	...
N-1	15	10
...	...	...
1	24	22
...	...	...
2	222	241
...	...	...
-	255	255

temporal $T_B(1)$

color patch index	target cam. #1	ref. cam. #0
-	0	0
-	1	0
-	2	1
-	3	1
1	4	1
-	5	2
-	6	2
-	7	3
N-1	8	3
...	...	...
2	12	6
...	...	...
0	255	201