Machine Vision for Automotive Displays Report

Optical Science Center
University of Arizona

May 7, 2014

1 Introduction

This project is developing a machine vision system for inspecting the display for an automative environment. The testing procedures including three main items:

1. Optics analysis: an analysis of the existing camera and lens choices to ensure they meet the resolution and field of view requirement and a basic description of choosing lenses for the future system that the screen dimensions may be different.
2. Test of illumination uniformity
3. Calibration methods for converting the RGB values of images obtained with the CCD camera to luminance measures in units of cd/m2.

2 Analysis of camera and lens

2.1 Theoretical analysis

1. Camera sensor format is the shape and size of the image sensor, for example,4/3"size is 17.3mm*13mm; 2/2"size is 8.8mm*6.6mm. As figure 1 shows.
2. FOV: the maximum angular size of the object as seen from the entrance pupil; the maximum angular size of the image as seen from the exit pupil.
3. Entrance pupil(EP) and exit pupil (XP) are the images of the stop in object space and image space.The aperture stop is the aperture in the system that limits the bundle of light that propagates through the system from the axial object point.
4. Effective focal length(EFL) of a system is determined from its front focal length f_F or rear focal length f'_R as $f_E = -f_F/n = f'_R/n'$
5. Aperture(F number): the image-space cone of the light for an object at infinity:

   $Fnumber = f_E/D_{EP}$

   D_{EP}=Diameter of the EP
6. Thin lens imaging:$z_o$is the object distance,$z_i$is the image distance.

   $$1/z_o + 1/z_i = 1/f$$
7. Transverse magnification or lateral magnification is the ratio of the image height h' to object height h, can also use the ratio of image distance to object distance to express.$m = h'/h = z_i/z_o$

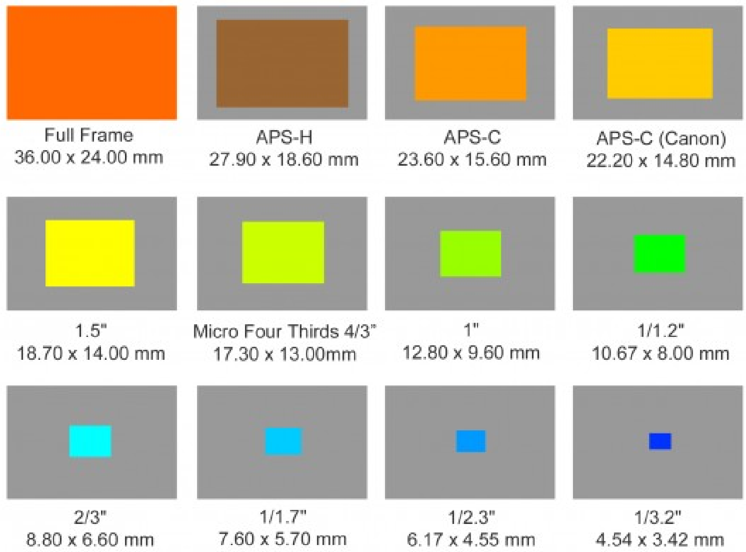

Figure 1: camera sensor size

2.2 Requirement

1. fullfill the working distance.
2. the image size of the test screen should be smaller than the sensor size.That is test screen size*m<sensor size.
3. in order to check each pixel of the display, the image size of each pixel should be larger than the pixel size of camera sensor. That is screen pixel size*m>sensor pixel size.

2.3 Analysis of existing system

The camera is Prosilica GT3300C with the following parameters:

Sensing Area, H x V (mm)	18.13 x 13.6,4/3"color
Pixels (H x V)	3296 x 2472
Pixel Size, H x V (um)	5.5 x 5.5

The optics is Kowa LM25JC10M, 2/3", 1.8/25mm C mount

Maximum Camera Sensor Format	2/3"
Picture size	6.6*8.8mm
Focal Length FL (mm)	25
Minimum focus distance	0.1m
Aperture (f number)	f1.8 - f16
Exit pupil	-84.4
Resolution	200lp/mm at center, 160lp/mm at corner

The size of the display screen is around 11cm*7cm.The working distance is 34.3cm; rough calculation:

$$1/z_0 + 1/z_i = 1/f$$

$$z_0 = 343mm, f = 25mm,$$

In this sense,

$$z_i = 27mm$$

so the magnification is

$$|m| = 27/343 = 0.0787$$

In this application , it must fullfill the size of object*m<size of sensor, that is

$$110 * 0.0787 = 8.58mm$$

$$70 * 0.0787 = 5.46mm$$

which even smaller than 2/3".

The pixel size of object *m > pixel size of sensor. For WVGA resolution which is 800*480, each display pixel size is around 0.1375mm*0.1458mm, so the image size of each pixel is 18um*11.477um, larger than the camera sensor pixel size.

In this sense, the existing optics works for the system.

3 Luminance uniformity

3.1 Process discription

1. Capture the image of display screen.
2. Read this RGB image in MATLAB as a 3-dimentional matrix
3. Extract from R,G,B channel to create 3 gray-scale image
4. Plot the gray scale image for each color channel which shows the luminance uniformity

3.2 Results

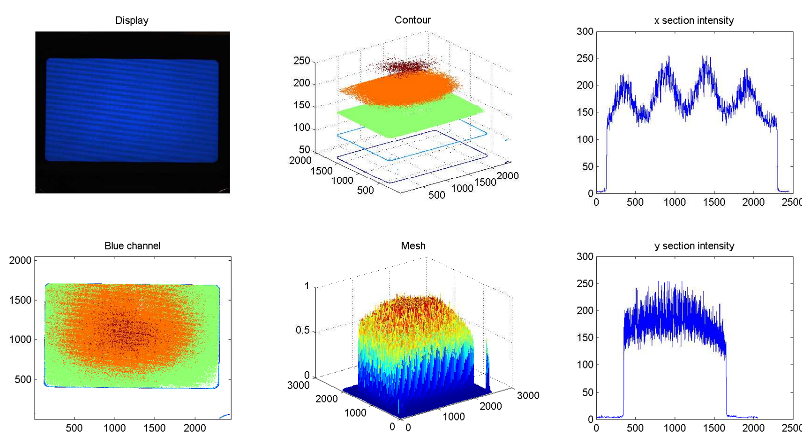

Figure 2: This blue Channel of luminance uniformity of blue screen

Figure 2shows the luminance uniformity of the blue channel for the blue display.
Figure 3shows the luminance uniformity of the green channel for the blue display.
Figure 4shows the luminance uniformity of the red channel for the blue display.

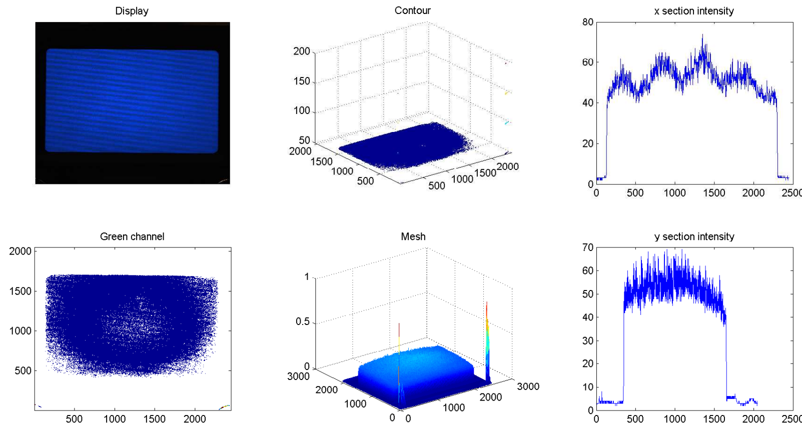

Figure 3: This green Channel of luminance uniformity of blue screen

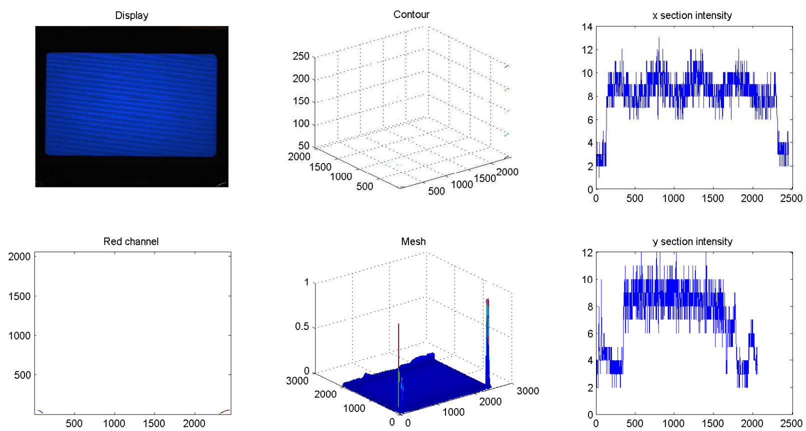

Figure 4: This red channel of luminance uniformity of red screen

3.3 MATLAB code

```
A=imread('C:\Users\xumoh_000\Desktop\Machine Vision\VP2-blue.bmp');%read image\\
Ablue=A(:,:,3);
Ared=A(:,:,1);
Agreen=A(:,:,2);%Extract from 3 color channel
[p,q]=size(Ared);
%Red channel
I=Ared;
figure();
subplot(2,3,1);
imshow(A);
title('Display');
subplot(2,3,4);
contour(flip(I));
title('Red channel');
subplot(2,3,2)
contour3(flip(I));
title('Contour');
subplot(2,3,5);
```

```
mesh(flip(im2double(I)));
title('Mesh');
subplot(2,3,3)
plot(I(p/2,:));
title('x section intensity');
subplot(2,3,6)
plot(I(:,q/2));
title('y section intensity');
%Green Channel
I=Agreen;
figure();
subplot(2,3,1);
imshow(A);
title('Display');
subplot(2,3,4);
contour(flip(I));
title('Green channel');
subplot(2,3,2)
contour3(flip(I));
title('Contour');
subplot(2,3,5);
mesh(flip(im2double(I)));
title('Mesh');
subplot(2,3,3)
plot(I(p/2,:));
title('x section intensity');
subplot(2,3,6)
plot(I(:,q/2));
title('y section intensity');
%Blue Channel
I=Ablue;
figure();
subplot(2,3,1);
imshow(A);
title('Display');
subplot(2,3,4);
contour(flip(I));
title('Blue channel');
subplot(2,3,2)
contour3(flip(I));
title('Contour');
subplot(2,3,5);
mesh(flip(im2double(I)));
title('Mesh');
subplot(2,3,3)
plot(I(p/2,:));
title('x section intensity');
subplot(2,3,6)
plot(I(:,q/2));
title('y section intensity');
```

4 Calibration

4.1 Discription

This section shows how to convert the RGB values of the display into luminance measures which capture by photometer. The basic idea is use a mask to sample the display screen as the Figure 5 shown. Firstly, find out the average red internsity ,average green internsity and average blue intensity for each sample area. Then use photometer to measure the luminance value for each sample area. The third, use multiple linear regression method to find out the coefficients R,G,B,D in

$$B * AvgBlue + R * AvgRed + G * AvgGreen + D = Photometervalue$$

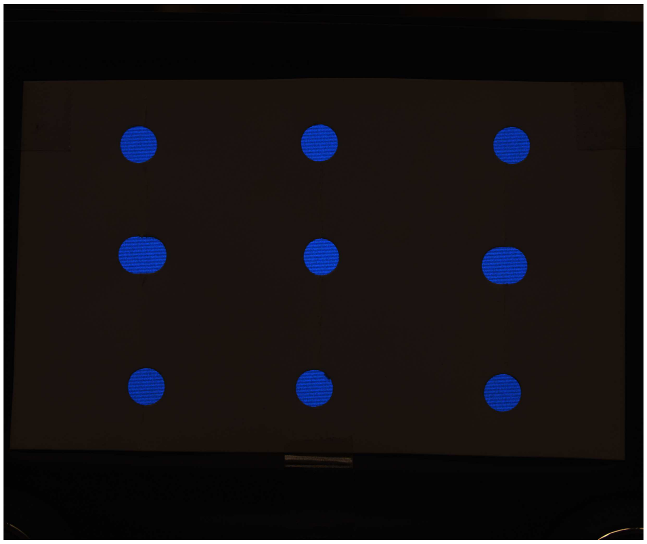

Figure 5: blue screen with sample

4.2 Results

The coefficients for the blue display are: $R = 3.2675$; $G = -1.0558$; $B = 0.6195$; $D = -23.0360$
That is

$$Photometervalue = 3.2675 * AvgRed - 1.0558 * AvgGreen + 0.6195 * AvgBlue - 23.0360$$

Figure 6 shows the fit result. red stars are the vaules measured by photometer, blue shows the fit result by the above formula.

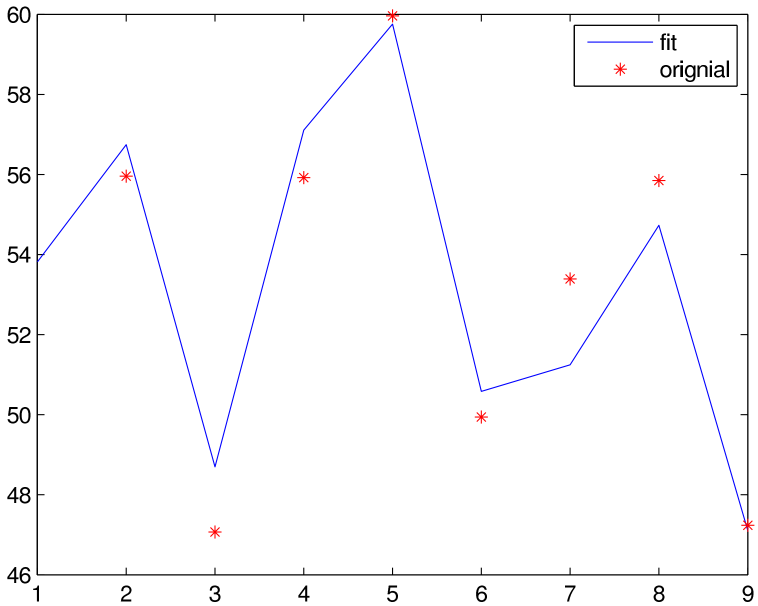

Figure 6: calibration

4.3 MATLAB code

```
BluePhoto=[54.46,55.92,53.39;55.96,59.96,55.85;47.07,49.94,47.24];
GreenPhoto=[371.3,374.0,349.7;409.1,409.8,387.8;312.0,343.7,321.2];
RedPhoto=[87.81,90.09,86.87;96.64,100.4,92.08;79.31,81.96,78.09];
WhitePhoto=[538.8,526.6,484.2;545.6,584.9,498.6;471.6,486.3,463.2];
A=imread('C:\Users\xumoh_000\Desktop\Machine Vision\blue-s.bmp');
imshow(A);
%A=double(A);
Ablue=A(:,:,3);
Ared=A(:,:,1);
Agreen=A(:,:,2);%Extract from 3 color channel
[h,l]=size(Ablue);
a=zeros(3,3);
AvgBlue=a;
AvgRed=a;
AvgGreen=a;
B=(im2uint8(Ablue≥50))/255;
Ablue=Ablue.*B;
Ared=Ared.*B;
Agreen=Agreen.*B;
D = bwdist(B);
DL = watershed(D);
bgm = DL;
figure, imagesc(bgm), title('Watershed ridge lines (bgm)')
figure()
for i=1:1:9
    Temp=bgm;
    Temp(Temp≠i)=0;
    Temp(Temp==i)=1;
    [p,q]=find(Temp==1);
    Mx=max(p);
    My=max(q);
    if Mx > 2000
          if My>2000
          a(3,3)=i;
          elseif My> l/2
             a(3,2)=i;
          else
              a(3,1)=i;
          end;
    elseif Mx >h/2
          if My>2000
          a(2,3)=i;
          elseif My> l/2
             a(2,2)=i;
          else
              a(2,1)=i;
          end;
    else
          if My>2000
          a(1,3)=i;
          elseif My> l/2
             a(1,2)=i;
          else
              a(1,1)=i;
          end;
    end;
    Tempblue=Ablue.*Temp;
    Tempred=Ared.*Temp;
    Tempgreen=Agreen.*Temp;
    index=(find(a==i));
    blue=sum(sum(Tempblue))/sum(sum(Tempblue≠0,2));
    AvgBlue(a==i)=blue;
    red=sum(sum(Tempred))/sum(sum(Tempred≠0,2));
    AvgRed(a==i)=red;
```

```
    green=sum(sum(Tempgreen))/sum(sum(Tempgreen≠0,2));
    AvgGreen(a==i)=green;
    subplot(3,3,index);
    imshow(Tempblue);
end
%then find R,G,B in B*AvgBlue+R*AvgRed+G*AvgGreen+D=Photometer value
AvgBlue=AvgBlue(:);%change the matrix into a vector
AvgRed=AvgRed(:);
AvgGreen=AvgGreen(:);
Avg=[AvgRed AvgGreen AvgBlue];

BluePhoto=BluePhoto(:); % change BluePhoto into RedPhoto,GreenPhoto,WhitePhoto accroding ...
    to the test pic.
F=BluePhoto;% change BluePhoto into RedPhoto,GreenPhoto,WhitePhoto accroding to the test pic.

%b=stepwisefit(Avg,F);
beta=[0 0 0 0];
betafit=nlinfit(Avg,F,'myfit',beta);
figure();
xa=1:1:9;
plot(xa,betafit(1)*AvgRed+betafit(2)*AvgGreen+betafit(3)*AvgBlue+betafit(4));hold on;
plot(xa,F,'*r');legend('fit','orignial');

%Check the regression method
AvgRegress=[ones(9,1) Avg];
[b,bint,r,rint,stats]=regress(F,AvgRegress);
```

function ‘myfit’ defined as:

```
function yhat=myfit(beta, x)
A=beta(1);
B=beta(2);
C=beta(3);
D=beta(4);
yhat=A.*x(:,1)+B.*x(:,2)+C.*x(:,3)+D;
```