צילום חישובי: תרגיל 1

API עבור הקוד שכתבנו:

הערה: על מנת להריץ את התכנה בשלמותה יש לרוקן את הזיפ לתוך תיקייה, להיכנס אלייה ולבצע את שורת הפקודה: make all. הפקודה תקמפל את התכנה dcraw ותמיר את התמונות שצירפנו לקבצי .tiff.
כדי להפעיל את הקוד על התמונות של הגיטרות יש למקם את קבצי ה*.CR2 בתקייה ולהריץ את הפקודה make guitar. ולאחר מכן לשנות את הריצה בקובץ הmain על מנת שיפעל עם המסגרת הנכונה לכרטיס האפור.

```
% Program running example
main

% White-Balance: Main program function
% This function produce a set of white-balanced images with different
% adaptation matrices according to the input images.
% inputs:
% graycardDir - path of graycard image file. *.tiff file
% withFlashDir - path of image shooted with flash. *.tiff file
% noFlashDir - path of image shooted without flash. *.tiff file
function [ ] = whiteBalance(graycardPath, withFlashPath, noFlashPath);

% This function reads .tiff image file, and normalize it
% input: img_file - .tiff filename
% output norm_img - normalized image in range [0 1]
function [ norm_img ] = tiff2double( img_file );

% This function get flash chromaticity according an image of a grey-card in the required
scene.
% input: image of grey card captured with flash as the only illuminant source.
%        image is *.tiff file format
%        x1, x2, y1, y2 - graycard borders
% output: flash chromaticity vector
function [ flashChroma ] = graycardBalance( img, x1, x2, y1, y2 );

% This function gets a mask of pixels influenced mostly by flash light.
% input: diff - difference image of flash-image and non-flash image
        lowerLimit, upperLimit - pixel value range
% output: binary mask of images in range (lowerLimit, upperLimit)
```

```
function [ mask ] = getMask( diff, lowerLimit, upperLimit);

% This function gets an image and chromaticity-vector,
% and apply disable image chromaticity.
% input: image, chromaticity vector
% output: image reduced chromaticity.
function [ output_image ] = disableChroma( img, chroma );

% Gamma correct input image using vision.GammaCorrector & imadjust
% functions.
% input: image
% output: gamma-corrected image.
function [ output_image ] = gammaCorrect( img );

% Apply adaptaion-matrix on image.
% Input: image, adaptation-matrix
% Output: adapted-image
function [ adaptImg ] = applyMatrix( img , adaptMat )
```

קבצים מצורפים:

main.m	white balance running example
whiteBalance.m	API function
tiff2double.m	API function
graycardBalance.m	API function
getMask.m	API function
disableChroma.m	API function
gammaCorrect.m	API function
Makefile	dcraw compiler & images generation in XYZ, RAW, sRGB

הקדמה:

בתרגיל התבקשנו לממש שיטה לביצוע white balance עבור תמונת קלט (נסמן בP), המוארת באמצעות מקור אור אחד לא ידוע, בהנתן 2 תמונות נוספות:

- האחת זהה לתמונת הקלט, אך בשונה ממנה צולמה עם פלאש בנוסף למקור האור המקורי(נסמן בF).
- השנייה היא תמונת כרטיס-אפור שצולמה בתאורת פלאש בלבד(נסמן בG).

הנחות:

1. בתמונה ללא הפלאש אנו מניחים כי קיים מקור אור יחיד.
2. הסצנה המוצגת בתמונה עם הפלאש וללא הפלאש היא אותה סצנה, ללא תזוזה.
3. נתוני המצלמה בתמונות אינם משתנים.

האלגוריתם:

- חילוץ נתוני הכרומטיות של הפלאש (flash chromaticity) מGm.
- חילוץ תרומת התאורה של הפלאש (נסמן בL) מF על ידי חישוב L=F-P.
- איתור פיקסלים שהושפעו מתאורת הפלאש במידה רצויה (לא בהירים או כהים מידי), נסמן pix.
- שימוש בנתוני הכרומטיות של הפלאש וL על מנת לחשב את R - ה"צבע הניטרלי" של הפיקסלים בpix.
- חילוץ נתוני הכרומטיות של מקור האור מP ע"י שימוש ב-R (בפיקסלים הנ"ל בלבד) ומיצוע.
- ביצוע איזון לבן בתמונת הקלט P על ידי חלוקת כל ערוץ צבע בערך המתאים מנתוני הכרומטיות של מקור האור,(לפי תאוריית הכרומטיות של וון קריס).
- ביצוע תיקון-gamma לתמונה.

<u>פירוט עבור האלגוריתם</u>:

<u>חילוץ נתוני הכרומטיות של הפלאש (flash chromaticity) מתמונת הכרטיס האפור</u>

ניקח חלון מתוך התמונה המכיל רק את הפיקסלים של הכרטיס האפור.

נבצע נרמול ע"י חישוב הממוצע של כל הפיקסלים בכל אחד מהערוצים, וחלוקה בערוץ האמצעי. כך נשיג את היחסים בין ערוצי הצבע שנוצרו כתוצאה מתאורת הפלאש. כלומר - את נתוני הכרומטיות של הפלאש.

ניתן לתאר זאת גם כמציאת ערכי f שיקרב את ערוצי הצבע של כל פיקסל על הכרטיס האפור לקבוע t כלשהו מהמשוואה:

$$\begin{pmatrix} t \\ t \\ t \end{pmatrix} = \begin{pmatrix} \frac{1}{fx} & 0 & 0 \\ 0 & \frac{1}{f_y} & 0 \\ 0 & 0 & \frac{1}{f_z} \end{pmatrix} \cdot \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix}$$

<u>חילוץ תרומת התאורה של הפלאש (חישוב L)</u>

על ידי חישוב L=F-P אנחנו מקבלים תמונה שמייצגת את תרומת התאורה של הפלאש לF. ניתן להתייחס לL כמעין תמונה של הסצינה שצולמה כביכול עם הפלאש כמקור האור היחיד.

<u>איתור פיקסלים שהושפעו מתאורת הפלאש בצורה רצויה</u>

נבחר את הפיקסלים בL בעלי ערך בטווח [0.4 0.6]. בחרנו בטווח זה משום שפיקסלים בעלי ערך נמוך מדיי אינם רלוונטיים מאחר ולא הופעו במידה רבה מתאורת הפלאש, ופיקסלים בעלי ערך גבוה מדיי אינם רלוונטיים מאחר והם מייצגים החזר אור מלא , השתקפות ישירה של הפלאש על עצמים מבריקים.

שימוש בנתוני הכרומטיות של הפלאש L על מנת לחשב את R - ה"צבע הניטרלי" של הפיקסלים שבחרנו

ערכי הפיקסלים ונתוני הכרומטיות של הפלאש נמצאים בידינו,
ע"י פתרון המשוואה

$$\begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} = \begin{pmatrix} \frac{1}{f_x} & 0 & 0 \\ 0 & \frac{1}{f_y} & 0 \\ 0 & 0 & \frac{1}{f_z} \end{pmatrix} \cdot \begin{pmatrix} pix_x \\ pix_y \\ pix_z \end{pmatrix}$$

עבור הפיקסלים בpix נוכל למצוא את R - הגוון המקורי של הפיקסלים בpix ללא השפעות כרומטיות מהפלאש.

חילוץ נתוני הכרומטיות של מקור האור מP ע"י שימוש ב-R (בפיקסלים הנ"ל בלבד) ומיצוע

$$\begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} = \begin{pmatrix} \frac{1}{L_x} & 0 & 0 \\ 0 & \frac{1}{L_y} & 0 \\ 0 & 0 & \frac{1}{L_z} \end{pmatrix} \cdot \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix}$$

נתון לנו R - הגוון המקורי של הפיקסלים בpix, ונתון לנו גם הערך שלהם לאחר ההשפעה של מקור האור. נמצא את הכרומטיות של מקור האור ע"י חילוץ הכרומטיות מהמשוואה הנ"ל (בפיקסלים הנ"ל בלבד).

ביצוע איזון לבן בתמונת הקלט P לפי תאוריית הכרומטיות של וון קריס

לאחר שמצאנו את הכרומטיות של מקור האור בשלב הקודם, בעזרת מערכת משוואות דומה לזו מהשלב הקודם, נמצא הפעם את R עבור כל פיקסל בתמונה, ונקבל את תמונת האיזון הלבן.
נעשה זאת באמצעות הפונקציה disableChroma עם נתוני הכרומטיות שמצאנו.

ביצוע תיקון-gamma

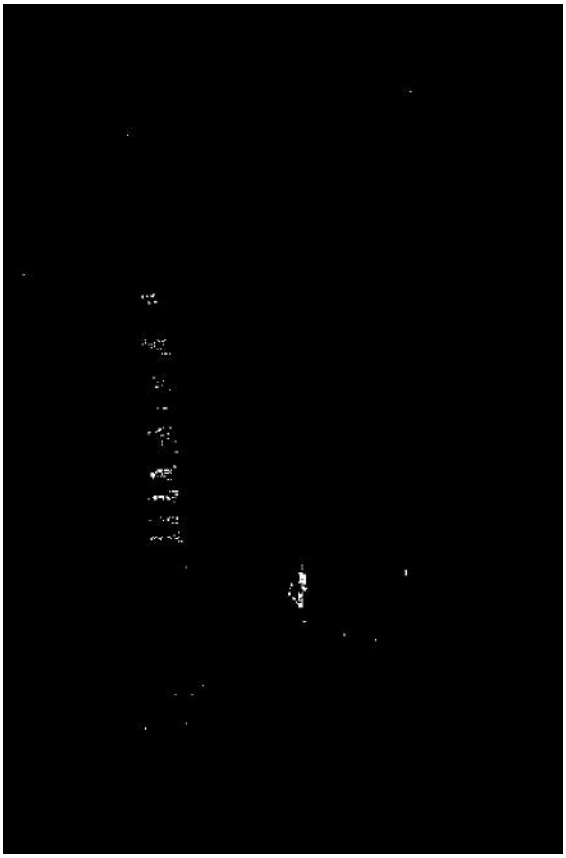
השתמשנו בפונקציות: vision.GammaCorrector & imadjust
ע"י ניסוי וטעייה מצאנו את ערכי הפרמטרים לביצוע בהירות ותיקוני gamma מתאימים.

<u>תוצאות והשוואה</u>:

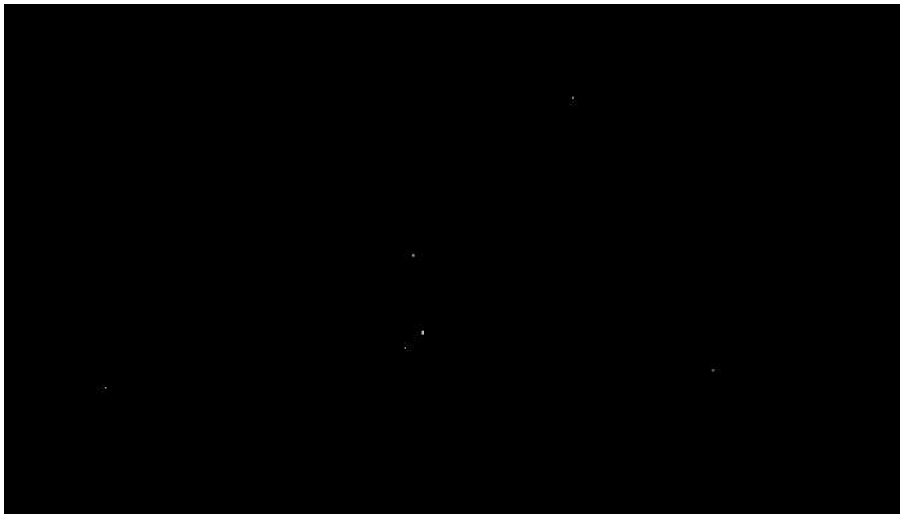
<u>בחירת הפיקסלים שהושפעו מתאורת הפלאש, בתמונת החיסור L:</u>

מצאנו כי מספר הפיקסלים שמושפעים בעיקר מהפלאש אינו גדול, כפי שניתן לראות בתמונות:

(a)



(b)



הפיקסלים שנבחרו בתמונת הגיטרות - (a)
הפיקסלים שנבחרו בתמונת הקלסרים - (b)

אך היה מספיק על מנת לחשב את הכרומטיות של מקור האור. ניסיונות להגדלת הטווח נתנו תוצאות טובות פחות כי כללו בחישוב פיקסלים עם ערכים לא אמינים, כפי שהוזכר לעיל.

<u>נסיונות מעבר בין מרחבי צבע</u>:

ניסינו להריץ את האלגוריתם תוך שימוש במרחבי צבע שונים בתמונת הפלט מdcraw. גילינו כי התוצאה הטובה ביותר היא שימוש בתמונת ה-RAW ללא המרה למרחב צבע ספציפי. התוצאה שהתקבלה היתה בעלת גווני הצבע הטובים ביותר, בהשוואה למרחבים האחרים. ההמרה למרחבי הצבע השונים התבצעה באמצעות יצירת קבצי ה-tiff באמצעות ה-flag המתאים בתוכנית dcraw:

RAW - 0, sRGB - 1, XYZ - 5

התוצאות שהתקבלו:

sRGB	XYZ

RAW

ניתן לראות כי התוצאות במרחבי הצבע sRGB וXYZ "סובלות" מנטייה קלה וחמורה לגוון צהוב בהתאמה. בעוד עבודה עם המרחב RAW כ (ללא המרה) נותנת תוצאה נקייה יותר.

נסיונות עבודה עם מטריצות אדפטציה שונות:

במהלך העבודה בדקנו את התוצאות גם תוך שימוש בטכניקות המרה שונות בין מרחבי צבע.
הטכניקות שניסינו הן:

1. vonKries
2. Bradford

התוצאות שהתקבלו:

vonKries	Bradford
	

לסיכום: התוצאה הטובה ביותר התקבלה ללא שימוש במטריצות אדפטציה.