

מעבדה בעיבוד אותות פיזיולוגיים

מעבדה מס' 2 - עיבוד תמונה

שמות המגישות/ים:

גיא פומרנץ 205375215

נועה ימין 208244913

מיכאל פולוניק 203833041

שם המדריך:

ליר כהן

תאריך הגשה:

13.12.2021

תוכן עניינים:

1. תקציר.....	3
2. ניסוי 1 – זיהוי מיקום אישונים.....	4
2.1. היפותזה.....	4
2.2. מתודולוגיה.....	4
2.3. תוצאות.....	4
2.4. מסקנות.....	12
3. ניסוי 2 – זיהוי תנועות רגליים.....	13
3.1. היפותזה.....	13
3.2. מתודולוגיה.....	13
3.3. תוצאות.....	13
3.4. מסקנות.....	16
4. ניסוי 3 – השפעת גודל המסנן על סינון רעש.....	18
4.1. היפותזה.....	18
4.2. מתודולוגיה.....	18
4.3. תוצאות.....	18
4.4. מסקנות.....	21
5. ניסוי 4 – זיהוי נטייה של טסיות דם.....	22
5.1. היפותזה.....	22
5.2. מתודולוגיה.....	22
5.3. תוצאות.....	22
5.4. מסקנות.....	26
6. מסקנות כלליות.....	28
7. ביבליוגרפיה.....	29
8. נספחים.....	30

1. תקציר

מטרת הניסוי היא הכרת מבנה התמונה והתכונות שלה הן בחום הזמן והן בתחום התדר, התנסות בביצוע פעולות על תמונה כגון סינון, פעולות מורפולוגיות, סגמנטציה וזיהוי גבולות. כמו כן, התנסות במיצוי מאפיינים של אובייקטים מתמונה ועיבוד וידאו.

בניסוי הראשון כתבנו אלגוריתם אשר מזהה מיקומי עיניים בהתאם למספרים השונים עליהם הנבדק הסתכל. ראינו כי האלגוריתם שכתבנו מאפשר חיזוי טוב של המספר עליו הנבדק הסתכל, אך בעל פער בזיהוי כאשר מדובר בקטעי המעבר בין הסגמנטים.

בניסוי השני ביצענו זיהוי לסמנים ידועים מראש כדי לקבל התרשמות על קצב הליכה ומרחק בעזרת ניתוח התמונה. ראינו כי לרוב הזיהוי צלח אך ישנם מספר פספוסים אשר היה ניתן לזהות את אותם פספוסים במידה והסמנים היו יותר גדולים כך שהם לא ימרחו תחת תנאי רזולוציה נמוכה.

בנוסף, ראינו כי לגודל המסנן קיימת השפעה ניכרת על אופי סינון של תמונה, הסקנו כי מדובר בהשפעה של גודל הסביבה של המסנן בהתאם לאופי הרעש וכן הראינו כי ההבדל בין הסינון של המסננים הינו מובהק.

כמו כן, בניתוח תמונת טיסות הדם ביצענו הפרדה המתבססת על זווית ההטייה של דמות, ראינו כי ניתן להשיג מאפיינים ומידע נוסף על כל דמות בנפרד בעזרת פונקציות מובנות וכך ניתן לשפר את יכולת הניתוח כאשר אנו רוצים למקד את העניין שלנו רק בחלק מהדמויות בעלות מכנה משותף.

לסיכום, בניסוי זה התנסינו בזיהוי מיקומי עיניים, תנועות רגליים ונטייה של טסיות דם זאת בשימוש באלגוריתמים שונים לעיבוד תמונה. כמו כן, בחנו את השפעת גדלי מסננים שונים על תוצאת סינון התמונה.

2. ניסוי 1 – זיהוי מיקום אישונים

2.1. היפותזה

בניסוי זה נבצע זיהוי של מיקומי עיניים בהתאם למיקומים שונים אליהם הנבדק יסתכל. נכתוב אלגוריתם אשר מזהה את מיקום העיניים בכל פעם ונכניס אליו סרטון על מנת שעבור כל פריים הוא יזהה את המיקום אליו העין הסתכלה. אנו מצפים כי האלגוריתם שנגדיר יזהה באופן טוב את המיקום אליו העין הסתכלה, אך צופים שתהיה בעיה בפריימים בהם העין נמצאת בשלבי ביניים של תזוזה, זאת כתוצאה ממיקום אשר לא מכויל וכן יתכן מטשטוש שינבע מתזוזת העין ואיכות לא טובה של המצלמה.

2.2. מתודולוגיה

כלים וחומרים:

מקבע לראש, מצלמה, gui לקבלת קובץ קבצי התמונות והסרטון.

מהלך ביצוע הניסוי:

וידאו שהאור בחדר כבוי למניעת רעש רשת, קבענו את הצוואר של הנבדק לשולחן הניסוי וקבענו את מיקום המצלמה מול העיניים שלו. לאחר מכן, פתחנו את קובץ המאטאלב משם vid_feed.m. על המסך הופיעו המספרים 1-4. צילמנו כ-12 תמונות כיול כאשר בכל פעם הנבדק הביט למספר אחד מבין המספרים 1-4 בסדר קבוע ושמרנו את התמונות בקובץ. אחר כך צילמנו את הנבדק בוידאו שאורכו 30 שניות כאשר בכל 2 שניות מיקום העין שלו השתנה לפי הסדר הבא:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2$

שמרנו את הסרטון ומיד בסיום השמירה הסרטון ההקלטה החלה באופן אוטומטי.

2.3. תוצאות

תשובה לשאלה 3.1:

כתבנו פונקציה שמקבלת פריים בודד ומחזירה את מיקום כל עין באותו פריים בתור מטריצה שהשורה הראשונה מסמלת את עין שמאל והשנייה את עין ימין. כמו כן, קואורדינטה X בטור ראשון וקואורדינטה Y בטור השני. קוד הפונקציה:

```
function [Eye_Pos]= EyePosition_FUNC(Frame)
% the function gets frame and returns the position of the Iris

function [Eye_Pos]= EyePosition_FUNC(Frame)
% the function gets frame and returns the position of the Iris

IMAGE = Frame;
N = length(IMAGE(1,:));
Pattern_R = IMAGE(:,ceil(N/2+10):end); %image for each eye
Pattern_L = IMAGE(:,1:ceil(N/2));
[centers_R, ~] = imfindcircles(Pattern_R,[6
45], 'ObjectPolarity', 'dark', 'Method', 'TwoStage'); %finding the Iris using find
circle function
```

```

[centers_L, ~] = imfindcircles(Pattern_L,[5
35], 'ObjectPolarity', 'dark', 'Method', 'TwoStage');

if isempty(centers_L) %if didn't find the Iris
    centers_L = [0 0];
else
    if length(centers_L(:,1)) > 1 %if find more than one
        GOODPosition_L = find(centers_L(:,2)==max(centers_L(:,2))); %row number of
nearest circle
        centers_L = centers_L(GOODPosition_L,:);
    end
end

if isempty(centers_R) %if didn't find the Iris
    centers_R = [0 0];
else
    if length(centers_R(:,1)) > 1 %if find more than one
        GOODPosition_R = find(centers_R(:,2)==min(centers_R(:,2))); %row number of
nearest circle
        centers_R = centers_R(GOODPosition_R,:);
    end
end

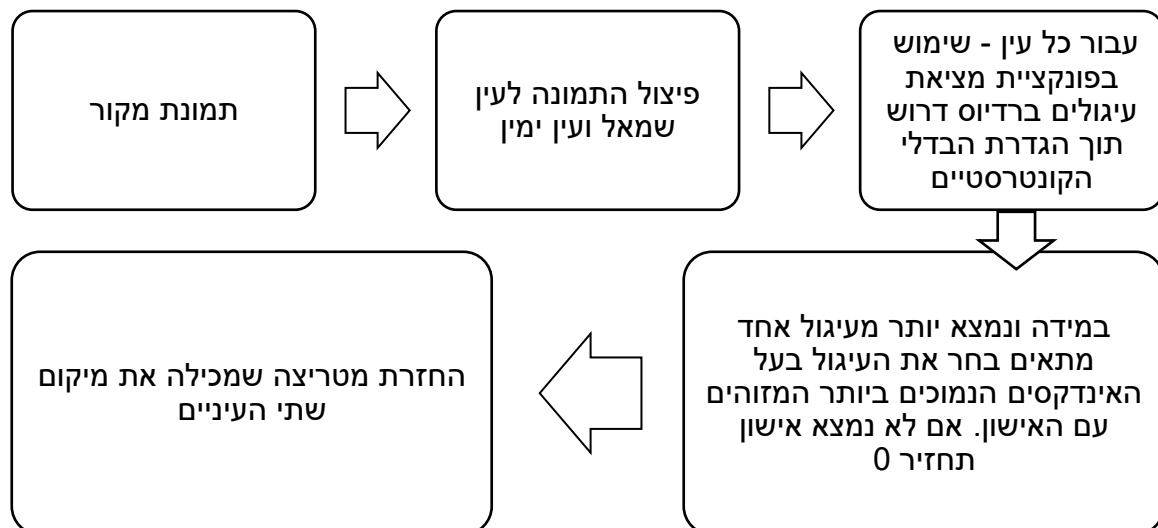
Eye_Pos = [round(centers_L) ; round(centers_R)]; %position of Irises

end

```

תשובה לשאלה 3.2 :

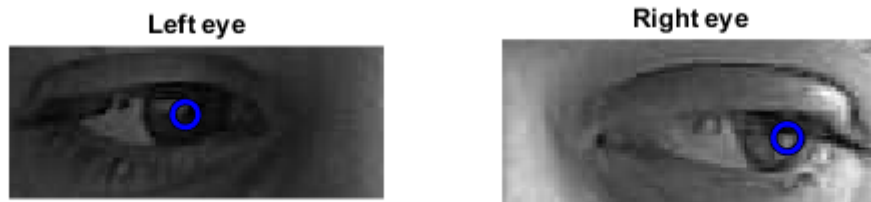
לצורך מציאת מיקום האישונים על גבי התמונה השתמשנו באלגוריתם הבא :



האלגוריתם בנוי כך שהפונקציה קיבלה תמונה המקור, פיצלה אותה לשתי תמונות אחת עבור כל עין. לאחר מכן, עבור כל עין, בוצע שימוש בפונקציית מציאת עיגולים ברדיוס דרוש תוך הגדרת הבדלי הקונטרסטים בין העין לסביבה הלבנה שלה. כמו כן, במידה ונמצא יותר מעיגול אחד מתאים בחר את העיגול בעל האינדקסים הנמוכים ביותר המזוהים עם האישון, כלומר הקרובים ביותר לצידה השמאלי של התמונה. לבסוף, הפונקציה מחזירה מטריצה שמכילה את מיקום שתי העיניים. כאשר השורה הראשונה שלה מסמלת את מיקום עין שמאל והשנייה את מיקום עין ימין. כמו כן, קואורדינטה X ממוקמת בטור

ראשון וקואורדינטה Y ממוקמת בטור השני. יתר על כן, במידה והפונקציה לא זיהתה אישון בכל אחת מהעיניים, היא מחזירה קואורדינטות אפס בתור שגיאה.

Iris found using the function



איור 1 - מיקום האישונים כפי שהתקבל בשימוש בפונקציה שנכתבה

באיור ניתן לראות את מיקום האישונים כפי שזוהה בשימוש בפונקציה שנכתבה. נבחין כי מיקום האישון עבור כל עין אכן זוהה בצורה טובה.

תשובה לשאלה 3.3 :

כתבנו פונקציה שמקבלת מטריצת פריים לכיול ועין ומחזירה את מיקום אחת העיניים כפי שהוגדרה בתור מטריצת מיקום של העין עבור כל פריים. קוד הפונקציה :

```
function [EyeCalib] = EyeCalibration_FUNC(CalibFrames,eye)
% the function gets a matrix of the calibration frames and returns
%a matrix of the location of eyes for each frame
%for one eye. if eye = 1 left eye, if eye = 2 right eye

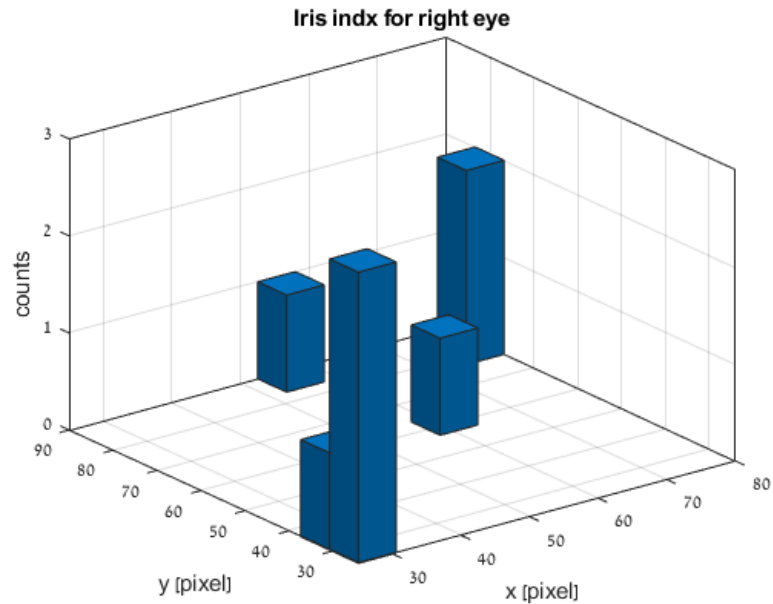
NUM = length(CalibFrames(1,:)); %number of calibration images
EyeCalib = zeros(NUM,8); %calibration place of eyes
i = 1;
for n = 1:4 %finding the place of Iris for eye
    for j = 1:NUM
        Frame = CalibFrames{n,j};
        Frame = rgb2gray(Frame); %color to BW photo
        Frame = im2double(Frame);
        Frame = Frame(230:277,155:390);

        Eye_Pos = EyePosition_FUNC(Frame); %finding eye position

        EyeCalib(j,i:i+1) = Eye_Pos(eye,:); %filling the calibration matrix with
the correct eye position
    end
    i = i + 2;
end
```

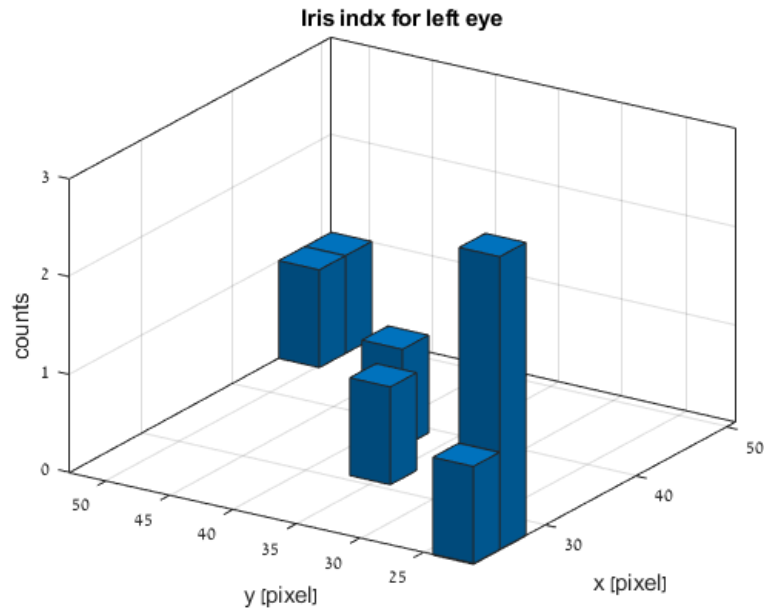
end

הפונקציה עושה שימוש בפונקציית מציאת מיקום עיניים שנכתבה בשאלה 3.2 מעלה.



איור 2 - היסטוגרמה עבור מיקומי העיניים ממטריצת הכיול עבור עין ימין

באיור ניתן לראות היסטוגרמה אשר מציגה את מיקום עין שמאל עבור כלל המספרים כפי שהתקבלה ממטריצת הכיול שחושבה באמצעות הפונקציה שכתבנו. נבחין כי אכן קיימים 4 פיזורים יחסית שונים של העיניים.



איור 3 - היסטוגרמה עבור מיקומי העיניים ממטריצת הכיול עבור עין שמאל

באיור ניתן לראות היסטוגרמה אשר מציגה את מיקום עין שמאל עבור כלל המספרים כפי שהתקבלה ממטריצת הכיול שחושבה באמצעות הפונקציה שכתבנו. נבחין כי גם פה אפשר להבחין ב-4 פיזורים יחסית שונים של העיניים.

תשובה לשאלה 3.4.1 :

כתבנו פונקציה שמקבלת את מטריצת הכיול ופריים הבדיקה ומחזירה את המספר עליו הסתכל הנבדק בפריים. קוד הפונקציה :

```
function [Eye_Look]= EyeLook_FUNC(Frame, EyeCalib)
% the function gets a frame and the calibration matrix and returns the
% number with the highest probability that the eye is looking at.
% by using the right eye coordinates

Eye_Pos = EyePosition_FUNC(Frame); %eye position for each eye in the frame
Right_Eye = Eye_Pos(2,:);

Mean_Number = mean(EyeCalib(:,1:2:end)); %mean for each number in X coordinate

%min distance from mean cali
NumLOOK_R = find(abs(Mean_Number-Right_Eye(1).*ones(1,4)) == min(abs(Mean_Number-
Right_Eye(1).*ones(1,4))));

Eye_Look = NumLOOK_R; %number looking at in frame

End
```

תשובה לשאלה 3.4.2 :

לצורך מציאת המספר עליו הנבדק הסתכל, השתמשנו בפונקציית הכיול לעין ימין, בהנחה כי כאשר הנבדק מסתכל על מספר הוא הסתכל עליו עם שתי העיניים. בשימוש בפונקציית מיקום העיניים, מצאנו את

מיקום העין הימנית הפריים הנבדק. לאחר מכן, בצענו ממוצע על מיקום ה-X בלבד של כל עמודת כיול המתאימה למספר אחר. כלומר, קיבלנו 4 מספרים אשר מהווים את ממוצע מיקום האישון עבור כל מספר עבור אינדקס X. לאחר מכן, חישבנו מה היא העמודה אשר ההפרש בין ציר ה-X של מיקום עין ימין שזוהתה מהפריים, למיקום בציר ה-X עבור כל ממוצע שקיבלנו לכל מספר, היה הקטן ביותר. מספר העמודה המתאימה הוחזרה בתור המספר עליו הנבדק הסתכל בפריים הנבדק. הסיבה למבנה אלגוריתם זה הוא משום שאנו מסיקים שעבור כל מספר אליו הנבדק יסתכל המיקום הינו יחסית זהה ולכן ביצוע ממוצע ייתן מיקום אינדקס מיטבי. כמו כן, מיקום האישון משתנה בעיקרו בציר ה-X ולכן ההשוואה בוצעה בציר זה ולא בציר Y אשר הינו קרוב מאוד עבור כלל המספרים.

טבלה 1 - המספר שהפונקציה זיהתה כתלות במספר עליו הנבדק הסתכל

מספר ידוע עליו היה המבט	המספר שהפונקציה זיהתה
1	1
2	2
3	3
4	4

בטבלה ניתן לראות כי הפונקציה אותה כתבנו זיהתה כראוי את כלל המספרים עליהם הנבדק הסתכל כנדרש. כלומר, הפונקציה זיהתה כראוי. בשלב זה בחנו את הסרטון אשר מכיל את תזוזות העיניים. יצרנו וקטור שורה אשר מכיל את הסימונים המתאימים בהתאם לסגמנטציה שבצענו באופן ידני מהסרטון. לאחר מכן, העברנו בכל פעם פריים בודד מהסרטון בפונקציה שכתבנו אשר מזהה את מיקום העין ובדקנו האם המספר שווה למספר הידוע מהסגמנטציה.

תשובה לשאלה 3.5 :

טבלה 2 – סגמנטציה של הסרטון המגדירה זמן מקטע של התבוננות על המספר

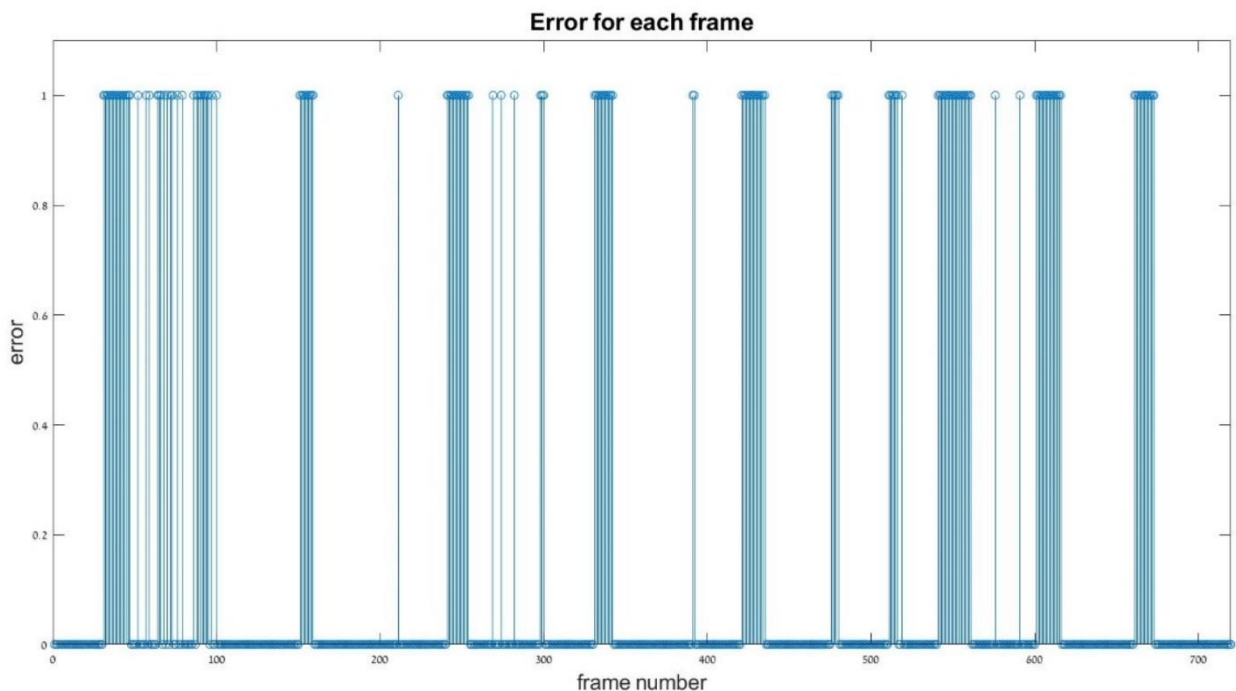
מספר	מספרי פריים	זמן מקטע (שניות)
1	1 – 30	0 – 1
2	31 – 90	1 – 3
3	91 – 150	3 – 5
2	151 – 210	5 – 7
3	211 – 240	7 – 8
4	241 – 300	8 – 10
3	301 – 330	10 – 11
2	331 – 390	11 – 13
1	391 – 420	13 – 14
2	421 – 480	14 – 16
1	481 – 510	16 – 17

17 – 18	511 – 540	2
18 – 20	541 – 600	3
20 – 22	601 – 660	1
22 – 24	661 – 720	2

בטבלה ניתן לראות את הסגמנטציה הידנית שבוצעה לסרטון שמגדירה את המספר עליו הנבדק הסתכל כתלות בזמני הסרטון.

תשובה לשאלה 3.6 :

לאחר הרצת הקוד התקבלו סך הכל $Error = 163$. כלומר, עבור 163 פריימים הזיהוי של מיקום האישון היה לא תקין. כלומר, 77.4% אחוזי הצלחה של האלגוריתם.



איור 4 - השגיאה עבור כל פריים כפי שהאלגוריתם זיהה

באיור ניתן לראות את השגיאה שהתקבלה בזיהוי המספר עליו הסתכל הנבדק כתלות במספר הפריים. נבחין כי הפיזור של השגיאות הוא בעיקר צפוף באזורים מסוימים. נבחן את האזורים הללו ואת הקשר שלהם להחלפה בין מקטעי התבוננות על מספרים. לאחר התבוננות בצברי השגיאה הרחבים, מצאנו שהפריימים בהם הם מתחילים הם: 31, 64, 86, 151, 241, 298, 331, 421, 476, 511, 541, 601, 661. מהתבוננות בטבלה מספר 2, ניתן להבחין כי ככל ערכים אלו מהווים את הפריים הראשון או נמצאים ממש בקרבת הפריים הראשון של סגמנטי מעבר בהתבוננויות על המספרים. כמו כן, שאר השגיאות התקבלו באופן מפוזר והן מעטות.

2.4. מסקנות

מהתוצאות ראינו כי לאחר בחינת האלגוריתם שיצרנו לזיהוי המספר עליו הסתכלה העין, קיבלנו כי כלל המספרים זוהה כנדרש בתהליך הכיול. מכך ניתן להסיק כי הפונקציות שכתבנו, הן עבור מציאת מיקום האישון, מציאת מטריצת הכיול וכן זיהוי המספר אליו הנבדק הסתכל הינו טוב. בנוסף, הבחנו כי עבור ההיסטוגרמות שהתקבלו למטריצת כיול לכל עין מכילות 4 פיזורים יחסית שונים של העיניים. זאת כמצופה משום שמדובר ב-4 מספרים שונים ולכן נצפה ל-4 מיקומי קואורדינטות שונים.

תשובה לשאלה 3.6 המשך :

כפי שציפינו, בתוצאות ראינו כי צברי השגיאה הרחבים של זיהוי האלגוריתם את מיקום התבוננות העין, מצאנו שערכי הפריימים המתאימים לתחילת סגמנט השגיאה הרחב, הם כולם ערכים אשר מהווים את הפריימים הראשון או נמצאים ממש בקרבת הפריימים הראשון של סגמנטי מעבר להתבוננויות על המספרים. האלגוריתם וכן תהליך הסגמנטציה שבוצעו לא לקחו בחשבון מצבי מעבר של העיניים. ממבנה האלגוריתם שנכתב, המספר עליו הסתכלה העין יוגדר בהתאם לנקודה הקרובה ביותר למספר אליו הסתכלה. מכך ניתן להסיק כי האלגוריתם שלנו בעל בעיה בזיהוי מקטעי מעבר, ויתכן כי גם תהליך ביצוע הסגמנטציה היה גס מדי ולא זיהה כראוי את הפריימים בהם בוצע המעבר להתבוננות על המספר השני. כמו כן, שאר השגיאות התקבלו באופן מפורז והן מעטות. ולכן נסיק כי מדובר באי זיהוי תקין של הפונקציות הנכתבות על שמיקום העין איננו במצב מעבר.

כמו כן, ראינו כי האלגוריתם שלנו זיהה כמעט 80% נכונה של המיקום אליו העין הסתכלה. ניתן לומר כי מדובר באחוז דיוק יחסית גבוה ולכן נסיק כי האלגוריתם שלנו, אומנם נתון לשיפור עבור קטעי המעבר לזיהוי תקין, אך עבור קטעים סטטיים שהם העין במיקום קבוע האלגוריתם זיהה בצורה טובה מאוד את הנדרש.

3. ניסוי 2 – זיהוי תנועות רגליים

3.1. היפותזה

בניסוי זה נזהה את תנועת הרגליים של אדם תוך כדי הליכה. הזיהוי יתבצע לפי סמנים. מכיוון שהסמנים הם שחורים והדמות לובשת בגדים בהירים יחסית, אנו מצפים לקבל אחוז גבוה של פריימים עם זיהוי מוצלח. אנו מצפים לראות תנועות יותר גבוה ככל שהסמן ממוקם יותר נמוך שכן ככל שהסמן יותר נמוך כך טווח התנועה גדל. נצפה כי איכות הזיהוי עבור הסמן בקרסול ובברך תהיה נמוכה מאשר של המותן והירך מהסיבה שהמותן והירך פחות זזים בפריים. לכן נצפה לפחות "המרחויות" של הסמן על גבי מספר פיקסלים בסמנים הגבוהים. כמו כן, אנו צופים כי המרחק הליכה אשר יזוהה יהיה יותר גבוה מהמרחק בפועל אשר עבר משום שחישוב המרחק מתבצע על פי המרחקים הקטנים בין כל פריים לפריים בסמן כך שיתקבל קריאה של קשת, בעוד שההליכה עצמה מתבצעת רק על המישור של ההליכון שמבחינתו בצילום הוא מהווה ישר.

3.2. מתודולוגיה

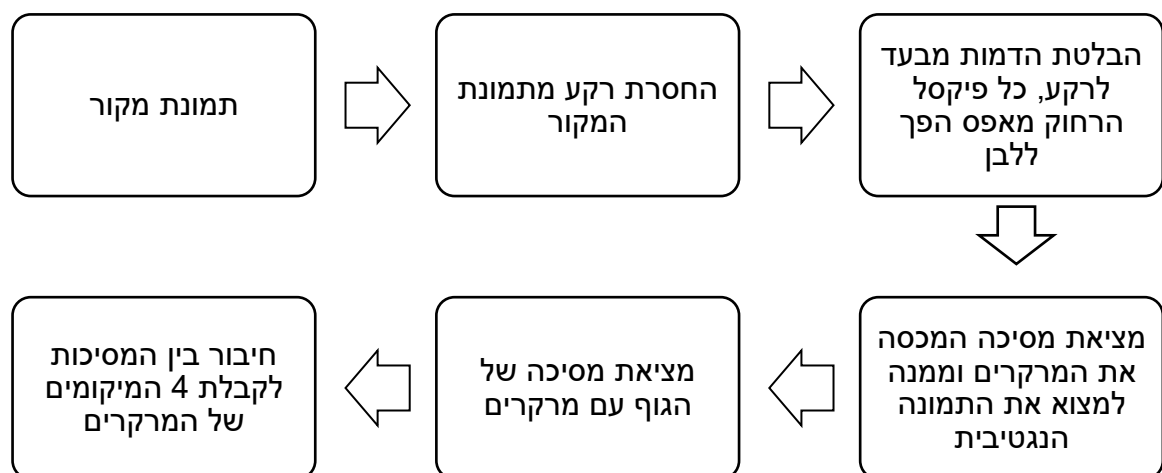
כלים וחומרים: 4 סמנים, מצלמה, הליכון, gui לקבלת קובץ avi של הסרטון.

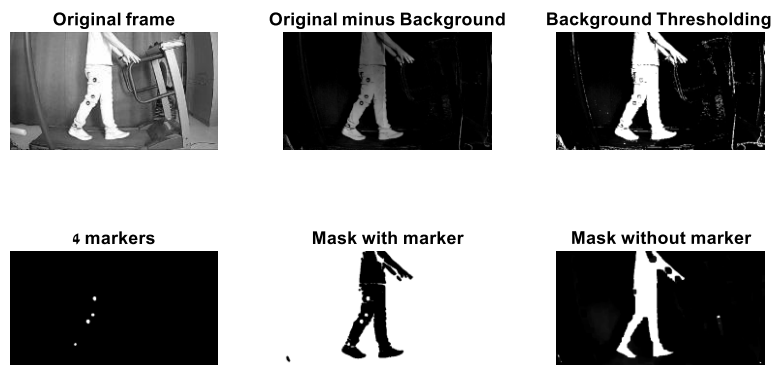
מהלך ביצוע הניסוי: מיקמנו את 4 הסמנים כך שיפנו כלפי המצלמה. סמן אחד על העקב, סמן נוסף מעל ומתחת לברך וסמן רביעי על המותן. הפעלנו את ההליכון על מהירות 2 קמ"ש והפעלנו את הסרטון למשך 60 שניות. לאחר הקלטת הסרטון נעבוד על כל פריים בנפרד ונזהה את 4 הסמנים לפי מיקום קואורדינטות (x,y) של כל סמן.

3.3. תוצאות

תשובה לשאלה 4.1.1:

לצורך מציאת המרקרים על גבי התמונה השתמשנו באלגוריתם הבא.



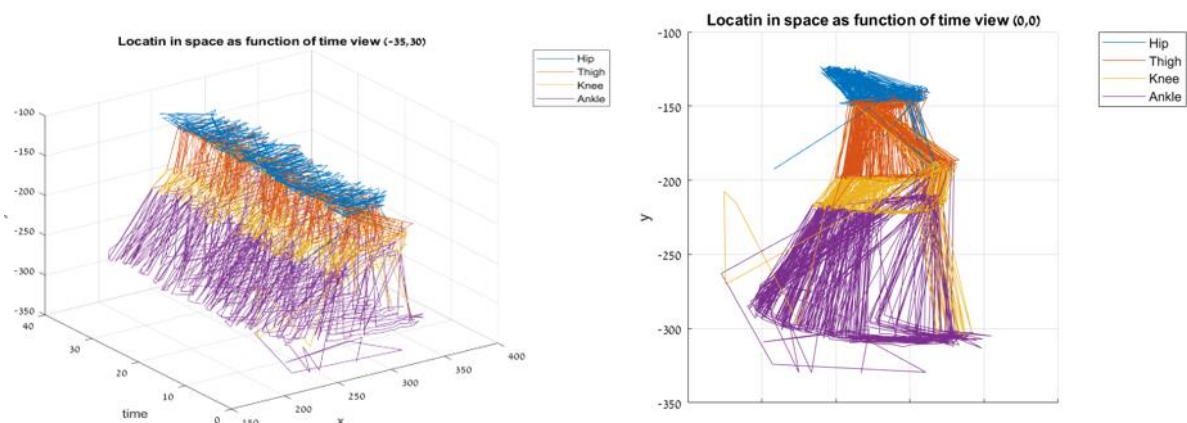


איור 5 - פריים לדוגמא בכל שלב באלגוריתם

האיור מתאר לנו בצורה טובה את מהלך האלגוריתם. תחילה בעזרת הרקע שצולם לפני שצולם הסרטון עשינו הבדלה ראשונית בין הדמות לרקע כדי ליצור מסיכה, חיפשנו את הפיקסלים שהערך שלהם בערך מוחלט הכי קרוב לאפס והשארנו אותם כמו שהם, וערכים מוחלטים הרחוקים מהאפס הוגדרו כ-1 וכך ניתן לראות בגרף מימין למעלה באיור. מגרף זה הוצאנו 2 מסיכות אשר מרכיבות לנו את הסמנים. מסיכה ראשונה בה רואים את הדמות אבל המרקרים לבנים, כלומר לא ניתן להבחין בהם ולמסכה זו נמצא את המראה הנגיבי שלה כמו שאנחנו רואים בתמונה מימין למטה באיור. מסיכה נוספת תופסת את הדמות בצורה יותר חדה ומשאירה את 4 הסמנים, ובעזרת קומבינציה של 2 מסיכות אלו אנחנו מקבלים את התמונה משמאל למטה אשר מכילה בה את ארבעת הסמנים. נוהה את הקואורדינטה של כל נקודה לבנה ונגדירה בתור הסמן.

תשובה לשאלה 4.1.2 :

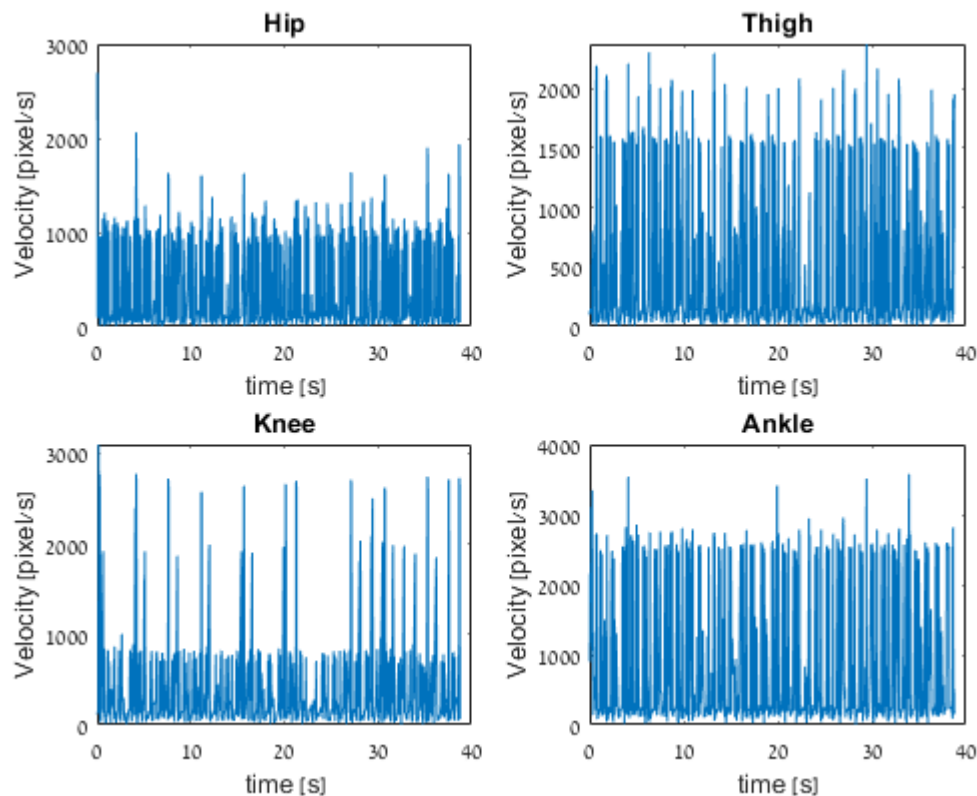
נסתכל על כל הזיהויים בגרף אחד באופן אחיד כדי לדעת אם ביצענו זיהוי מספיק טוב או לא.



איור 6 - מיקום כל סמן במרחב בהתאם לזמן הפריים

באיור ניתן לראות את המיקום של כל סמן במרחב בהשתנות לפי הזמן. בכחול ניתן לראות את הסמן שעל המותן, בכתום הסמן על הירך, בצהוב הסמן שמתחת לברך ובסגול הסמן שעל הקרסול. מהאיורים אפשר לראות באופן כללי את איכות הזיהוי, וניתן להבחין בתנועה כללית שמראה על הליכה.

תשובה לשאלה 4.2 :



איור 7 - מהירות הסמן ביחידות של פיקסל לשנייה

בין כל שני פריימים צמודים מצאנו את המרחק שהסמן עשה בתנועה כאשר הנחנו כי הסמן נע אך ורק על מישור התמונה. ובעזרת הפרש הזמן והפרש המרחק מצאנו את המהירות בכל זמן. קודם כל בולט כי התנועה של הקרסול הייתה הכי גבוה לאורך כל הזמן (גרף מימין למטה באיור). אחריו המהירויות הכי גבוהות היו בברך (גרף שמאל למטה), והסמן בעל המהירויות הנמוכות ביותר הוא הסמן על המותן. נשים לב כי רוב הסמנים מראים גרף די אחיד מלבד הסמן של הברך, ניתן לראות פיקים באופן די מחזורי בהם הברך נעה יותר מהר ביחס לשאר הזמן.

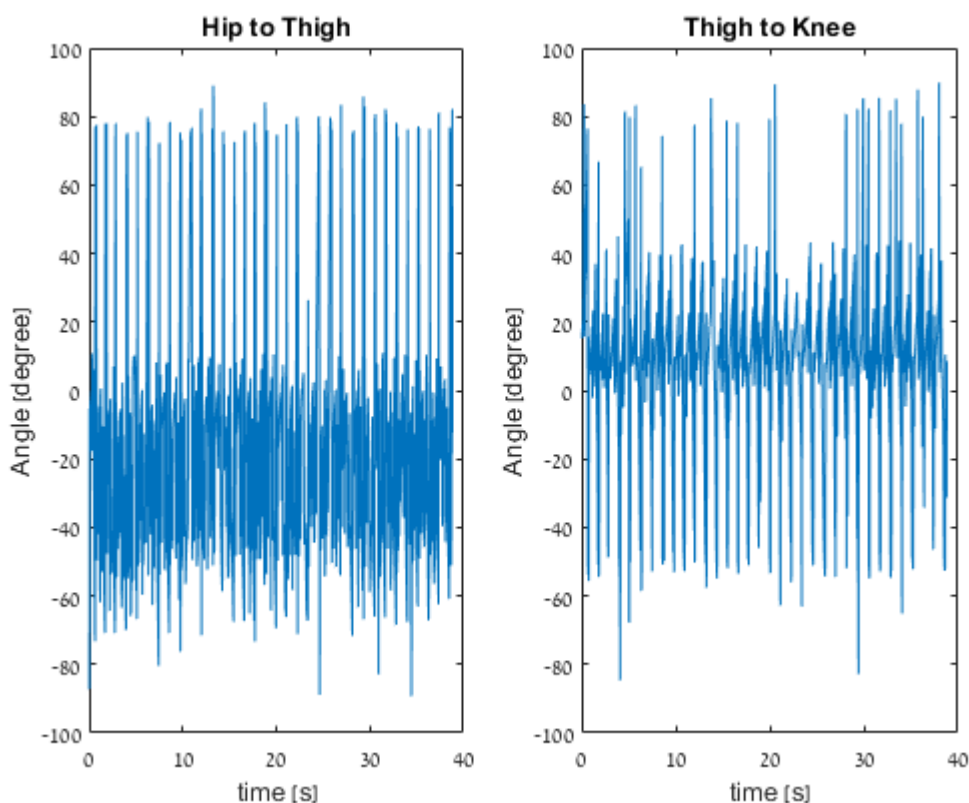
תשובה לשאלה 4.3 :

המרחק הכולל אשר עשה הסמן על הקרסול חושב לפי מרחק בין הפיקסלים. מכיוון שכל כדור הוא בקוטר של ס"מ וכל כדור לוקח כ-14 פיקסלים, נבצע המרה של מרחק מפיקסל לס"מ על ידי חלוקה של המרחק הכולל ב-14. לאחר מכן המרנו את ערך זה למטרים והתקבל

מרחק הליכה : 50.835 מטר.

תשובה לשאלות 4.4 + 4.5 :

הנחנו כי הדמות מוצבת בזווית של 180 מעלות כאשר וקטור אנכי כלפי מטה מוגדר כישר בעל זווית אפס. על סמך מערכת צירים זו מצאנו את הזווית הנוצרת מהישר המחבר את הירך והמותן לבין בציר המרכזי. בנוסף מצאנו את הזווית הנוצרת מהישר המחבר את סמן הירך לסמן הברך ביחס לציר המרכזי. הזוויות שהתקבלו הן:



איור 8 - זווית לפי זמן בין הירך והמותן לציר המרכזי משמאל, ובין הברך והירך לציר המרכזי בכגרף מימין

דבר ראשון ניתן לראות כי קיבלנו אות בעל אופי מחזורי עם פיקים אשר חדים יותר בצד אחד שעל גבי הציר האנכי. בגרף מצד שמאל אנחנו רואים את הזווית הנוצרת מהירך והמותן לציר המרכזי, אפשר לראות שאנחנו מקבלים את הזוויות הגבוהות ביותר בכיוון החיובי וכי המחזוריות היא כזאת שכל מחזור מגיעים כמעט לזווית חיובית של 80 מעלות ובצד השלילי מגיעים למינוס 50 מעלות. מצד שני בין הברך לירך מרבית הזוויות הן שליליות, מידי פעם יש קפיצות בעלות זווית גבוה חיובית אך האופי המחזורי מראה על זווית שלילית של 50 מעלות.

3.4. מסקנות

בניסוי זה ביצענו זיהוי ל-4 סמנים הממוקמים במקומות קבועים על הרגל בכל פריים, כפי שניתן לראות בתוצאות הצלחנו לבצע זיהוי מספק כדי לקבל את מרבית הפריימים כהצלחות, לכן אפשר להסיק מכאן שהאלגוריתם שבנינו מספיק טוב לזיהוי אך כנראה לא ספציפי מספיק כך שהתפספסו פריימים בהם התרחשה תופעה של אליאסינג או שאיכות הרזולוציה לא הייתה מספיק טובה. כאשר מסתכלים על כל הזיהויים בבת אחת אפשר לראות שחלק מהזיהויים שביצענו הינם שגויים, כלומר הסמן הנכון לא זוהה אלא אזור מוצל אחר בפריים. כדי לשפר את היכולת זיהוי נסיק כי אם נבצע את מהלך הניסוי אם לבוש יותר ניטרלי, שיכיל בתוכו פחות קפלים ומקומות אשר עלולים להזדהות כסמן אז נקבל זיהוי יותר טוב

בניתוח סרטון. מהסיבה הזאת גם קיבלנו כי מרחק ההליכה היה גבוה יותר מהמצופה, עבור סרטון של 40 שניות והליכה בקצב של 2 קמ"ש ההתקדמות היא 22 מטר ואנחנו קיבלנו מרחק כפול מזה. נשים לב כי צפינו לקבל מרחק הליכה גדול יותר מאשר מה שאמור להתקבל בפועל, לכן מהבחינה הזו כן תאמנו לציפיות, אבל אצלנו התקבלה חריגה של יותר מפי 2 מהמרחק בפועל שלא ניתן ליחס אותה אך ורק לעובדה כי חישוב המרחק נעשה בין פריים לפריים. נסיק כי גורם נוסף לשגיאה בחישוב מרחק ההליכה הוא העובדה שישנם פריימים אשר דילגנו עליהם, כתוצאה מאיכות ירודה מדי, לכן אנו לא רואים את התנועה במלואה ומידי פעם יש מספר קפיצות, אותן קפיצות משפיעות על איכות המדידה שלנו. כמו כן כפי שצינו ישנם זיהויים שגויים אשר השפיעו על חישוב מרחק ההליכה, לכן נסיק שעבור מערכת המזוהה טוב את הסמנים היינו מקבלים מרחק הליכה יותר קרוב. מהתבוננות על המהירויות של כל סמן, אפשר להבין יותר טוב את הסיבה מדוע היה קשה יותר לאתר את הסמנים הנמוכים, וזה מכיוון שהם היו בעלי המהירות הזמנית הגבוה ביותר, מהירות זו כפי שאמרנו פגעה ברזולוציה של אותם סמנים כך שלפעמים הם לא זוהו. נסיק מכאן שאם קצב ההליכה היה יותר נמוך אולי לא הייתה מתקבלת תופעת אליאסינג אשר השפיעה על יכולת הזיהוי.

4. ניסוי 3 – השפעת גודל המסנן על סינון רעש

4.1. היפותזה

בניסוי נבחן את השפעת גדלי מסנני חציון שונים על הסרת רעש מלח פלפל מתמונה. כמו כן, נבחן האם ההבדל בין סינון המסננים הינו מובהק לאחר 100 חזרות. אנו מצפים שההבדל ביניהם יהיה מובהק משום שתהיה השפעה של גודל הסביבה על אופן סינון הרעש. משום שגודל מסנן שונה מגדיר גודל סביבה שונה במהלך הסינון [1].

4.2. מתודולוגיה

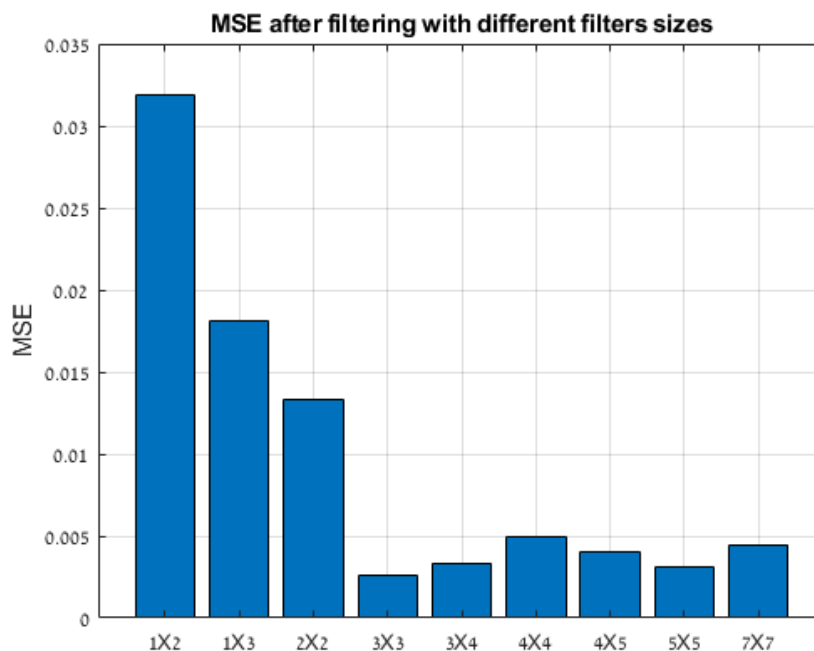
כלים וחומרים:

מאטאלב.

מהלך ביצוע הניסוי:

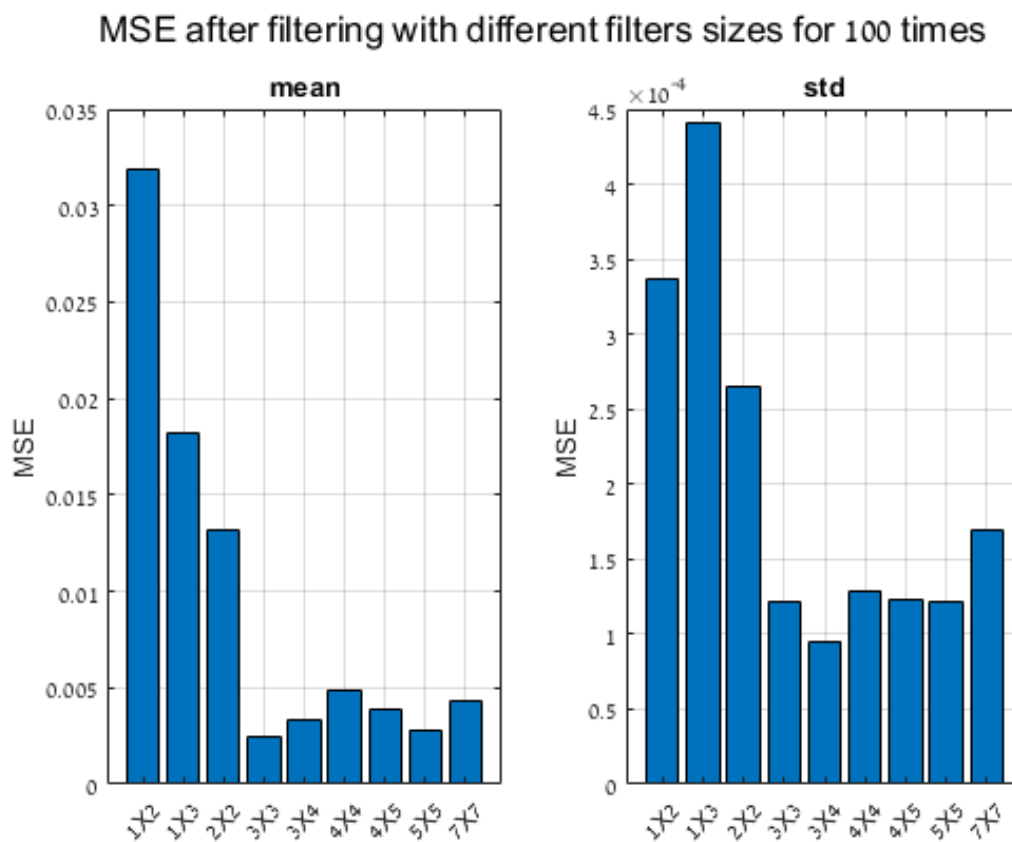
בסימלוציה לקחנו תמונה של ירקות והוספנו עליה רעש מלח פלפל של 20%. לאחר מכן הגדרנו 9 מסנני חציון בעלי גודל שונה וסיננו באמצעותם את התמונה. לאחר מכן, חישבנו את שגיאת ה-MSE לאחר סינון אחד, ולאחר 100 חזרות. תוך בחינת ממוצע ה-MSE וסטיית התקן שלה. כמו כן, בחנו באמצעות מבחן אנובה האם ההבדל בין המסננים הינו מובהק.

4.3. תוצאות



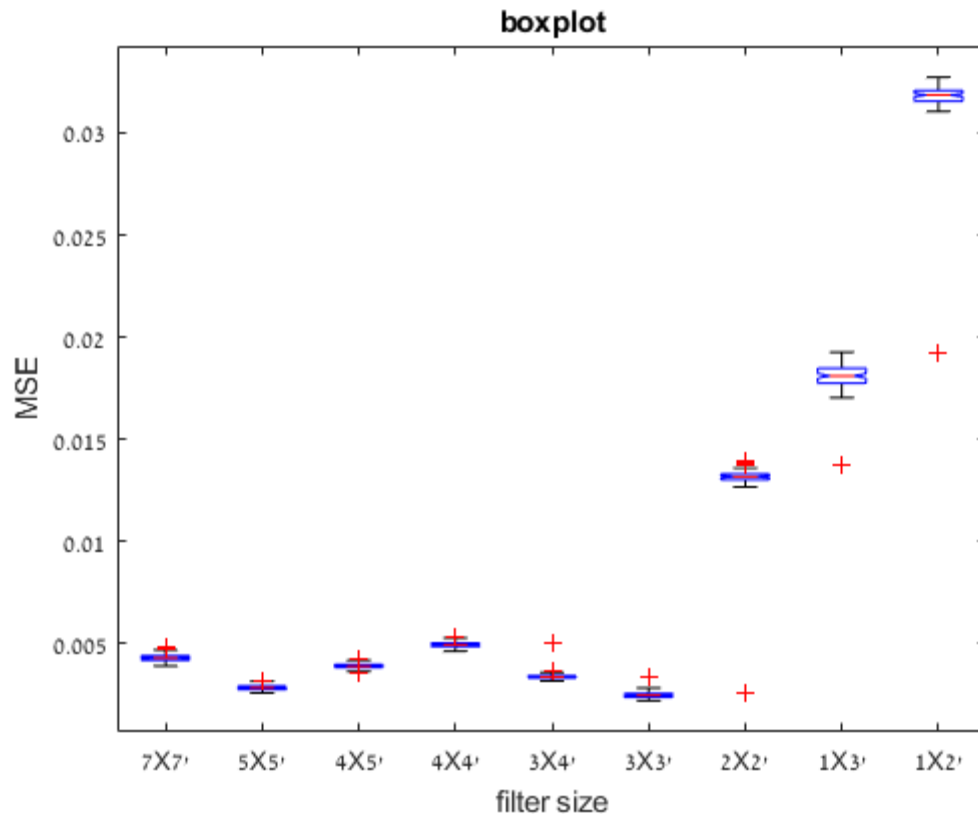
איור 9 - MSE כתלות בגודל המסנן

באיור ניתן לראות את השגיאה הריבועית כתלות בגודל מסנן החציון. נבחין השגיאה המינימלית התקבלה עבור מסנן בגודל 3×3 והמקסימלית עבור מסנן 1×2 .



איור 10 - ממוצע וסטיית תקן של MSE לאחר חזרה על הניסוי 100 פעמים

באיור ניתן לראות את הממוצע והסטיית תקן של השגיאה הריבועית כתלות בגודל מסנן החציון, כאשר הניסוי בוצע 100 פעמים. נבחין השגיאה המינימלית התקבלה גם פה עבור מסנן בגודל 3×3 והמקסימלית עבור מסנן 1×2 . כמו כן, סטיית התקן הקטנה ביותר הייתה עבור מסנן בגודל 3×4 ולאחריה מסנן בגודל 3×3 .



איור 11 - boxplot של השגיאה הריבועית כתוצאה מסינון במסננים שונים

באיור ניתן לראות את השגיאה הריבועית של כל מסנן כאשר כלל הנתונים של ה-100 חזרות נלקחו בחשבון. נבחי כי באופן ברור לעין השגיאה של המסנן בגודל 3x3 היא הקטנה ביותר.

ANOVA Table					
Source	SS	df	MS	F	Prob>F
Groups	0.07548	8	0.00943	25332.49	0
Error	0.00035	891	0		
Total	0.07583	899			

איור 12 - מבחן anova

באיור ניתן לראות את תוצאות מבחן ה-anova שבוצע. מבחן סטטיסטי למציאת שוני בין קבוצות. כאשר הקבוצות שהוגדרו פה הן גדלי המסננים והערכים שהושוו הן השגיאה הריבועית. ערך p שהתקבל הוא $p = 0 < \alpha = 0.05$. לכן, עבור מבחן מובהקות של 5% ואף של פחות מכך התקבל כי באופן מובהק הקבוצות שונות אלו מאלו. כלומר, קיים שוני מובהק בין השגיאה הריבועית כתוצאה מסינון במסננים השונים.

4.4. מסקנות

בתוצאות ראינו שלאחר חזרה אחת, השגיאה המינימלית התקבלה עבור מסנן בגודל 3×3 והמקסימלית עבור מסנן 1×2 . ככל הנראה מסנן זה היה קטן מדי ולא אפשר שימוש בערכי סביבה מספיקים למציאת ערך החציון הדרוש על מנת להסיר את הרעש מהתמונה. כמו כן, נסיק כי הסביבה של מסנן 3×3 הייתה אידיאלית לסינון מיטבי של התמונה מבין המסננים הנבחרים. כאשר הניסוי בוצע 100 פעמים, הבחנו בתוצאות שהשגיאה המינימלית התקבלה גם פה עבור מסנן בגודל 3×3 והמקסימלית עבור מסנן 1×2 . כמו כן, סטיית התקן הקטנה ביותר הייתה עבור מסנן בגודל 3×4 ולאחריה מסנן בגודל 3×3 . כלומר, ראינו כי גם לאחר ביצוע הסינון כ-100 פעמים נוספות עדיין התקבל שהמסנן עם השגיאה הנמוכה ביותר הוא בגודל 3×3 . גם סטיית התקן שלו לא מאוד גדולה ולכן נסיק כי הוא מבצע סינון טוב. יתר על כן, באמצעות מבחן אנובה ראינו כי ההבדל בין הסינון במסננים השונים הינו מובהק. כלומר, באופן מובהק סטטיסטית נעדיף להשתמש במסנן בגודל 3×3 משום שככל הנראה הוא בעל הסביבה האידיאלית לסינון חציון מיטבי של התמונה להפחתת רעש מלח פלפל. ולכן ניתן להסיק כי מסנן זה הוא הטוב ביותר מבין המסננים שנבחנו.

5. ניסוי 4 – זיהוי נטייה של טסיות דם

5.1. היפותזה

בחלק זה של הניסוי אנחנו נמצא את טסיות הדם אשר נוטות ימינה. מהתבוננות בתמונת הטסיות אפשר לראות שהרקע אינו אחיד לכן ניתן לצפות כי שערך הרקע ושימוש ברקע המשווער לצורך מציאת המסכה יאפשר לנו לקבל תוצאות יותר טובות. נצפה כי בצורה הזאת נקבל היסטוגרמה יותר החלטית שיהיה יותר קל לעבוד איתה ולהחליט מה רקע ומה טסית דם. כמו כן, בגלל שיש טסיות אשר נוגעות אחת בשנייה אנו מניחים כי נמצא מספר טסיות אשר יחשבו כאחת.

5.2. מתודולוגיה

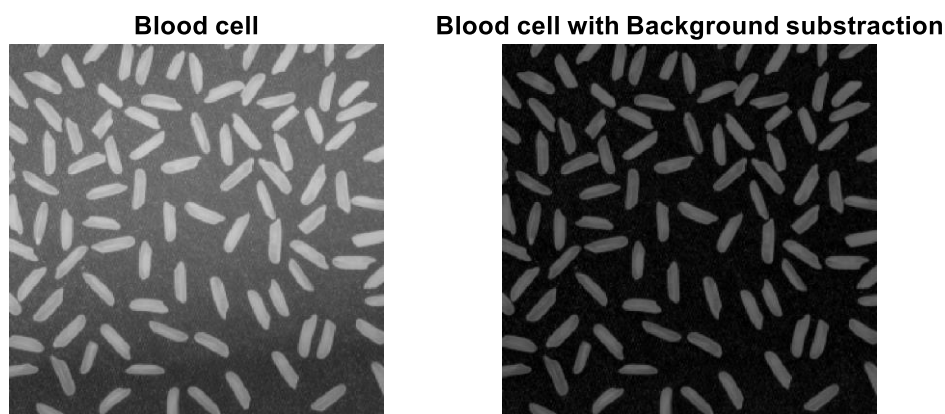
כלים וחומרים: Matlab, R2021b ; תצלום טסיות דם

מהלך ביצוע הניסוי: נכתוב אלגוריתם להבדלת טסיות דם נוטות ימינה משאר טסיות הדם. נבצע סינון אשר יתבסס על ההיסטוגרמה של התמונה תחת ההנחה כי טסיות הדם בהירות לעומת הרקע הכהה. לאחר ניקוי רעשים נמצא מסכה תואמת שתבדיל בין טסיות הדם לרקע. עם המסכה הבינארית נשערך את מספר טסיות הדם ולאחר מכן נבדיל בין הטסיות שנוטות ימינה לעומת הטסיות שנוטות שמאלה.

5.3. תוצאות

תשובה לשאלה 1.1: העלו את הקובץ

טענו את תמונת טסיות הדם אשר נמצאת בתיקיית מטלב, ביצענו שערך של רקע התמונה והחסרנו אותה מהתמונה המקורית. שערך הרקע יעשה בעזרת פעולת פתיחה open, קומבינציה של 2 פעולות מורפולוגיות חיסור והרחבה (Erosion & Dilation), קודם מבצעים החסרה ולאחר מכן מבצעים הרחבה בעזרת אותה תבנית [1]. נבחר תבנית בעלת שטח גדול יותר מרוחב טסיות הדם פעולת ההחסרה תחמיץ את הטסיות לכן נקבל פיקסלים כהים היכן שטסיות הדם קיימות בדומה לשאר הרקע, ופעולת ההרחבה תיתן לנו רקע בעל מבנה אחיד.

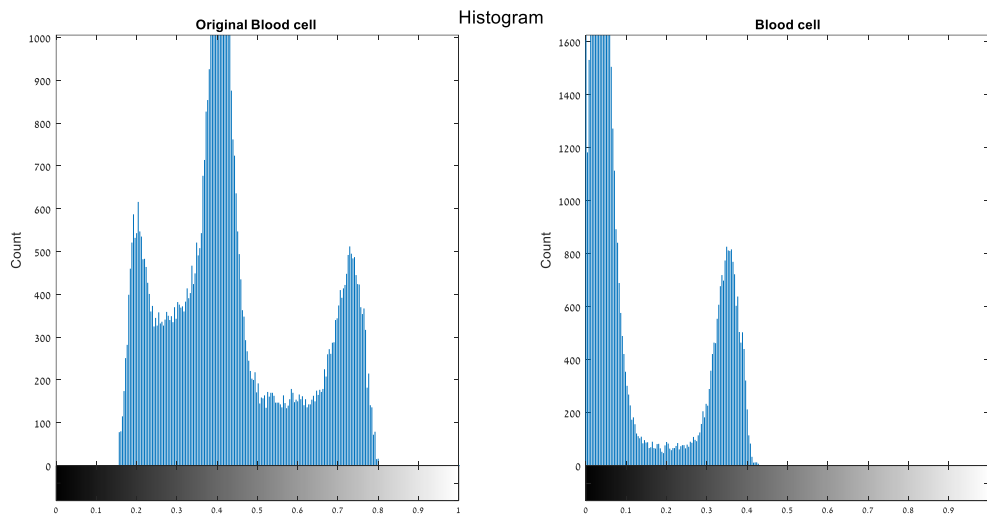


איור 13- משמאל תמונה מקורית של טסיות הדם, מימין טסיות הדם לאחר החסרת הרקע המשווער

באיור מצד שמאל אפשר לראות את התמונה המקורית, נשים לב כי החלק התחתון כהה מאוד ביחס לשאר התמונה ובעיה זו נפתרה כפי שצוין למעלה. מימין ניתן לראות את תמונת טסיות הדם לאחר החסרת הרקע, אפשר לראות את ההבדל החד בין הטסיות לבין הרקע.

תשובה לשאלה 1.2 : הציגו את ההיסטוגרמה רמות האפור של התמונה

מצאנו את ההיסטוגרמה של שתי התמונות :



איור 14- משמאל ההיסטוגרמה של התמונה המקורית, מימין ההיסטוגרמה של התמונה אחר החסרת הרקע

משמאל אנחנו רואים ההיסטוגרמה בעלת שלושה פיקים מרכזיים , ניתן להבדיל כי התחום הכי ימני בטווח ערכים של בין 0.7-0.8 זהו התחום בו נמצאות מרבית תסיות הדם, אבל ישנם מספר תסיות אשר מתפרשות על הטווח החופף עם הרקע. לעומת זאת, מימין ניתן לראות רק שני פיקים מרכזיים וניתן להניח כי הפיק הימני יכול את תסיות הדם בהבדלה מהרקע.

תשובה לשאלה 1.3 : נסו למצוא סף יחיד שמבדיל בין התסיות ובין הרקע. הציגו תמונת שחור לבן של המסכה הטובה ביותר שהצלחתם למצוא

בעזרת ההיסטוגרמה נבחר את סף החלטה, על סמך הנתונים נבחר סף החלטה בגודל של 0.2. כל הערכים השווים או קטנים מ-0.2 יהיו שחורים (אפס), וכל הערכים הגדולים מ-0.2 יהיו לבנים (אחד).



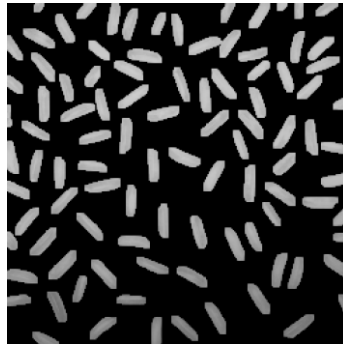
איור 15 - המסכה הטובה ביותר בעזרת סף החלטה

אם נשווה לאיור הראשון בסעיף נוכל לראות כי אכן קיבלנו מסיכה טובה וברורה שמכילה את כל הטסיות. בעזרת פעולת הפתיחה "open" הבלטנו את הטסיות וניקינו את הרעשים.

תשובה לשאלה 1.4 : צרו מבנה בצורת דיסק (עם פקודת strel ובעזרתו העלימו את הטסיות לקבלת הרקע. מהו רדיוס הדיסק שבחרתם? מעבר לניסוי וטעייה, הסבירו איך יש לבחור את הרדיוס על פי גודל הטסית

כדי להיפטר מהרעשים ברקע השתמשנו בפעולת הפתיחה "open" כך כל רעש אשר יהיה בעל קוטר הקטן מהאלמנט ימחק. לכן בחרנו באלמנט בצורת דיסק, כדי לבחור את גודל הדיסק ראינו קודם מהו הרוחב של טסית דם אחת, ראינו כי טסיות הדם בעלות גודל של 8 פיקסלים או יותר. מכאן שעלינו לבחור מבנה בעל רדיוס קטן מ-4, זאת משום שפונקציית strel בונה דיסק בעל קוטר של $2r + 1$ כלומר 9 פיקסלים. לכן, בחרנו מבנה דיסק בעל רדיוס של 3 פיקסלים. כפי שניתן לראות באיור 14 קיבלנו תמונה המראה רק את טסיות הדם וכל הרקע שחור. נבחר אלמנט בצורת דיסק בעזרת פונקציית strel כך שיהיה בעל רדיוס של 3 פיקסלים.

Only the Blood Cells with no background

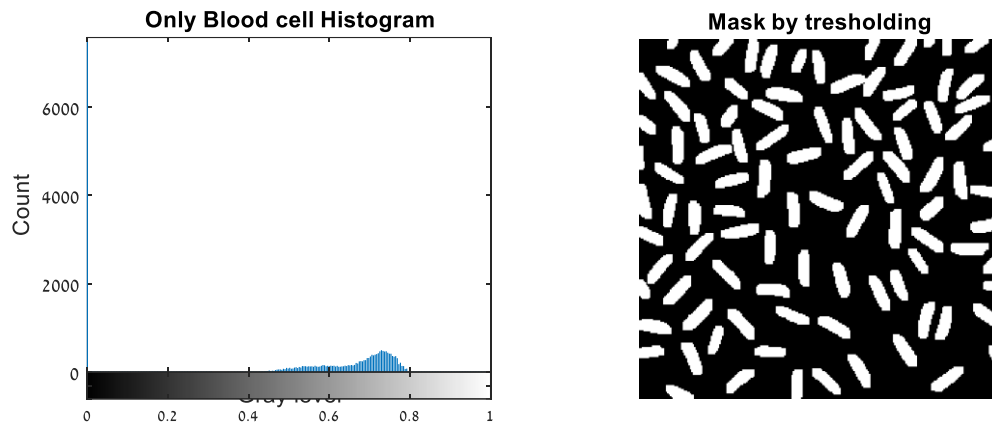


איור 16- טסיות הדם ללא רקע

ניתן לראות בתמונה רק את טסיות הדם וכל שאר הפיקסלים מוגדרים כשחור (אפס).

תשובה לשאלה 1.5 : חזרו על סעיפים 1.2 + 1.3 עבור התמונה לאחר הפחתת רקע שהתקבלה בסעיף 1.4 בצעו זיהוי גבולות של הטסיות בשתי דרכים.

עם התמונה באיור 15 חזרנו על השלבים הקודמים באלגוריתם :



איור 17 - משמאל היסטוגרמת התמונה ללא הרקע, מימין המסיכה שהתקבלה בעזרת ההיסטוגרמה

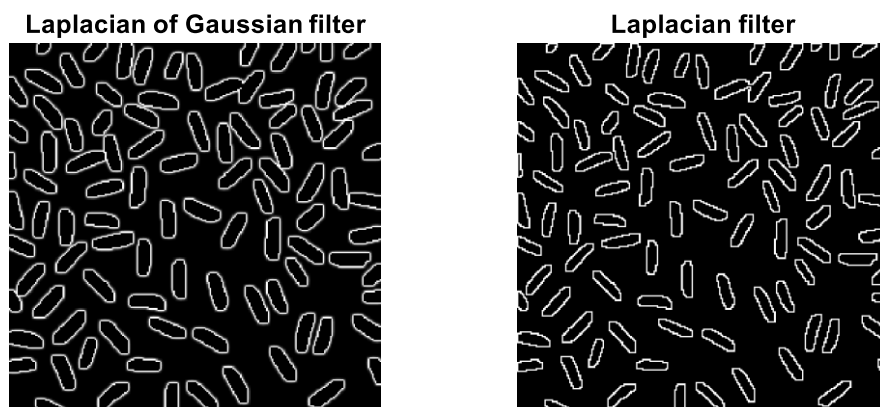
בחלק השמאלי של האיור אפשר לראות את ההיסטוגרמה של התמונה לאחר הכפלתה במסכה. קל לראות כי מרבית הערכים בתמונה כיהא לרקע הם עכשיו אפס ובחלק הימני של ההיסטוגרמה אפשר לראות רק את הפיקסלים המייצגים את תסיות הדם. מצד ימין ניתן לראות את המסכה החדשה, אפשר לראות כי היא ללא רעש ויש הבחנה ברורה של התסיות.

למסכה שקיבלנו ביצענו זיהוי גבולות בשתי שיטות שונות, פעם אחת בעזרת מסנן לפלסיאן – Laplacian, ופעם נוספת בעזרת לפלסיאן של גאוסיאן – Laplacian of Gaussian. למסנן גאוסיאן נשתמש בפילטר הבא [1]:

$$W = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

בשימוש בלפלסיאן של גאוסיאן, הגאוסיאן עוזר בהחלקת התמונה וכתוצאה מכך מפחית רעשים, ולאחר מכן מחשבים את הלפלסיאן עבור התמונה אשר מניב תמונה עם קצוות כפולים [1].

Blood Cells Border



איור 18 - זיהוי גבולות של תסיות הדם

באיור ניתן לראות זיהוי גבולות של הטסיות בשתי דרכים. מימן ניתן לראות את זיהוי הגבולות בעזרת לפלאסיאן בלבד ומשמאל לפלאסיאן של גאוסיאן. אפשר לראות כי בשימוש בגאוסיאן קיבלנו תמונה בעלת גבולות יותר צרים.

תשובה לשאלה 1.6 : הניחו כי התמונה היא יחידת שטח. שערכו מספר טסיות ליחידת שטח מתוך המסכה שיצרתם.

ביצענו שערוך למספר טסיות הדם בתמונה בעזרת פונקציית bwlable, הצגנו את שערוך טסיות הדם והתקבל :

Estimation of Blood Cell Number is : 97

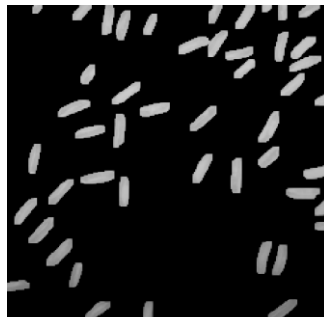
תשובה לשאלה 1.7 : בעזרת הפקודות bwlable ו regionprops מצאו רק את הטסיות הנוטות ימינה (/) וצרו מסכה המכילה רק אותן.

כדי להבדיל בין טסיות הדם הנוטות ימינה לשאר טסיות הדם, נעזרנו בפונקציית regionprops אשר מספקת לנו מידע על תכונות של דמויות בתמונה, כדי להבדיל בין הדמויות נעזרנו במשתנה אשר התקבל מפונקציית bwlable. כך, קיבלנו עבור כל דמות מספר מאפיינים, מיצינו את המאפיין אשר מתאר את הזווית נטייה של כל דמות regionprops.Orientation ולכל טסיות הדם אשר נוטות זווית הנטייה שלהן נמצאת בטווח של $0 < Orientation < 90$ הגדרנו את הפיקסלים כלבן (אחד), וכל מה שלא התאים הגדרנו אותם כשחור (אפס).

תשובה לשאלה 1.8 : הציגו את הטסיות הנוטות ימינה בלבד

עם המסכה מצאנו רק את הטסיות שנוטות ימינה בעזרת הכפלה בין המסכה לתמונה המקורית.

Only right tilted Blood Cells



איור 19- טסיות הדם שפונות ימינה בלבד

באיור ניתן לראות את התוצאה הסופית לאחר ניתוח תמונת הטסיות הדם.

5.4. מסקנות

כאשר מסתכלים על התמונה המקורית ולצידה התמונה לאחר החסרת הרקע אשר לראות באופן ברור את ההבדל בין ההיסטוגרמות, חיזוק לכך ניתן לראות בהיסטוגרמה של כל תמונה בה לתמונה המקורית יש 3 פיקס מרכזיים בעוד לתמונה לאחר החסרת שערוך של הרקע יש היסטוגרמה בעלת 2 פיקס מרכזיים.

מכאן, נסיק שצעד זה הועיל לנו לקבל הבדלה טובה של טסיות הדם לעומת הרקע, והועיל גם להמשך שלבי האלגוריתם.

באיור 16 אפשר לראות מסכה לאחר שחזרנו על צעדים קודמים באלגוריתם. נשים לב כי ברמת האיכות אין הרבה הבדל בין המסכה באיור 16 לעומת המסכה באיור 15, לכן נסיק כי שלב זה באלגוריתם לא מועיל בהרבה ובעיקר מוסיף לנו זמן חישוב ועיבוד.

תשובה לשאלה 1.5: בצעו זיהוי גבולות של הטסיות בשתי דרכים. הסבירו את ההבדלים ביניהם.

מהתבוננות באיור 18 אפשר לראות כי מסנן גאוסיאן בנוי כך שהוא מנמיך את התדרים המרחביים במרכז (התדרים הנמוכים), בצורה כזאת הוא מנחית את האזורים בהם אין שינוי כמו הרקע או שטח הטסית, ומצד שני מבליט את השינויים אשר מתרחשים בגבולות [1].

באיור אפשר לראות שהשימוש בלפלאסיאן של גאוסיאן הניב לנו תוצאה בה גבולות הטסיות יותר רחבים מאשר שימוש רק בגאוסיאן. הגבולות היותר צרים אשר ניתן לראות בשימוש בגאוסיאן בלבד עוזרים לנו להבדיל עוד יותר בין הטסיות הקרובות אשר חולקות גבול משותף, או אפילו הגבול הטשטש ביניהן מרוב הקרבה. לכן, אנו מסיקים כי השימוש במסנן גאוסיאן יותר טוב לצורך מציאת הגבולות בזיהוי טסיות דם.

מהתבוננות באיור 19 אפשר לראות שאכן קיבלנו אך ורק טסיות דם הנוטות ימינה ובנוסף נבחין כי יש בערך כחצי ממספר הטסיות הכולל לכן ניתן להניח כי התהליך נעשה בצורה טובה ונסיק כי האלגוריתם יעיל.

6. מסקנות כלליות

מהניסוי הראשון ניתן להסיק כי על מנת לאפשר זיהוי בדיוק של בקירוב 100% של מיקומי עיניים, יש לקחת בחשבון באלגוריתם את שלבי המעבר בין הסגמנטים וכן למצוא דרך מדויקת יותר למצוא את נקודת המעבר ברזולוציה של פריים ולא של שניות.

בניסוי השני כאשר ביצענו זיהוי של תנועת הרגל ראינו את ההשפעה הרבה שיש לתופעת האליאסינג ומהירות ההליכה על התמונה אשר מתקבלת מכל פריים. הסקנו כי שימוש בסמנים גדולים על גבי רקע מבדיל תשפר יותר טוב את יכולת הזיהוי אבל עדין גם לאלגוריתם פשוט יש יכולת זיהוי מספקת.

בנוסף, אנו מסיקים כי כאשר נרצה לסנן תמונה, יש לקחת בחשבון את גודל המסנן תוך בחינה של גדלים שונים לקבלת הסינון המיטבי של הרעש מהתמונה.

כאשר ביצענו ניתוח על טסיות הדם האדומות, ראינו את ההשפעה של הרקע על התמונה והסקנו כי על מנת לבצע ניתוח טוב והבדלה טובה בין דמות לרקע, כדאי להשתמש בשערוך רקע לקבלת מסיכה יותר החלטית שלא מתבססת אך ורק על רמות האפור.

7. ביבליוגרפיה

- [1] R. E. Woods, S. L. Eddins, and R. C. Gonzalez, "Digital image processing using MATLAB," 2009.

```

%exp 2 - image processing
%% Q-1-----

%part 1
clear all;clc;

%1.1
Bcr= imread('rice.png');%Bc for Blood Cell r for Raw
Bcr = double(Bcr)/255; %scale as double between 0-1
% estimating the bckground for better masking
se = strel('disk',10);
Bco = imopen(Bcr,se);
Bc = Bcr - Bco;%subtracting the background

figure
subplot(121)
imshow(Bcr)
title('Blood cell')
subplot(122)
imshow(Bc)
title('Blood cell with Background subtraction')

%1.2
figure
sgtitle('Histogram')
subplot(121)
imhist(Bcr)
title('Original Blood cell ')
xlabel('Gray level')
ylabel('Count')
subplot(122)
imhist(Bc)
title('Blood cell')
xlabel('Gray level')
ylabel('Count')

%1.3
% treshholding
Bc_MASK= Bc;
Bc_MASK(find(Bc<=0.15))=0;
Bc_MASK(find(Bc>0.15))=1;

figure
imshow(Bc_MASK)
title('mask after treshholding')

%1.4
element = strel('disk',3); %element to do imopen
Bc_cleanMask = imopen(Bc_MASK,element);
Only_Bc = Bc_cleanMask.*Bcr;

figure
imshow(Only_Bc)
title('Only the Blood Cells with no background')

```

```

% %1.5
% %1.52

%1.53
Bc_MASK2= Bc_cleanMask;
Bc_MASK2(find(Bc_cleanMask<0.25))=0;
Bc_MASK2(find(Bc_cleanMask>0.25))=1;

figure
subplot(121)
imhist(Only_Bc)
title('Only Blood cell Histogram')
xlabel('Gray level')
ylabel('Count')
subplot(122)
imshow(Bc_MASK2)
title('Mask by thresholding')

LoG = fspecial('log'); %creating LoG filter
BcAfterLoG = imfilter(Bc_MASK2,LoG); %filteing the image

figure
sgtitle('Blood Cells Border')
subplot(121)
imshow(BcAfterLoG)
title('Laplacian of Gaussian filter')

LaplaceFilter = [-1 -1 -1 ; -1 8 -1 ; -1 -1 -1]; %laplacian matrix
BcLapla = imfilter(Bc_MASK2,LaplaceFilter);
subplot(122)
imshow(BcLapla)
title('Laplacian filter')

%1.6
Label = bwlabel(Bc_MASK2);
Estimate_Bc_Num = max(max(Label)); % finding the highest element number wich will
be thr estimation
disp('Estimation of Blood Cell Number is:')
disp([Estimate_Bc_Num])
Orientation=cell2mat(struct2cell(regionprops(Label,'Orientation')));
[~,I] = find(Orientation>0 & Orientation<90); % Marking all the Bc that are tilted
between 0 - 90 degree

%1.8
% we will make a loop to flag the element number providing by the bwlabel
% with the a scalar which is equal to 1 + num of element found
Only_Right_Mask = zeros(size(Label));
for i = 1:length(I)

    Only_Right_Mask(find(Label==I(i)))=1;

end
Only_Right = Only_Right_Mask.*Bcr;
figure
imshow(Only_Right)
title('Only right tilted Blood Cells')

%% Q-2-----

```

```

%% 2.1
clear all

SPveg = imread('Nvegetables.tif'); %loading Image
SPveg = im2double(SPveg);
% [mv,nv] = size(SPveg); %size image
veg = imread('vegetables.tif');
veg = im2double(veg);

m = [7 5 4 4 3 3 2 1 1]; %size of filter
n = [7 5 5 4 4 3 2 3 2];

restoredefaultpath
rehash toolboxcache

Imafter77 = CleanSP(SPveg, 'Median', m(1), n(1)); %filtering the image using median
filter with different sizes
Imafter55 = CleanSP(SPveg, 'Median', m(2), n(2));
Imafter45 = CleanSP(SPveg, 'Median', m(3), n(3));
Imafter44 = CleanSP(SPveg, 'Median', m(4), n(4));
Imafter34 = CleanSP(SPveg, 'Median', m(5), n(5));
Imafter33 = CleanSP(SPveg, 'Median', m(6), n(6));
Imafter22 = CleanSP(SPveg, 'Median', m(7), n(7));
Imafter13 = CleanSP(SPveg, 'Median', m(8), n(8));
Imafter12 = CleanSP(SPveg, 'Median', m(9), n(9));

MSE_77 = mse(veg, Imafter77); %MSE for each filtering
MSE_55 = mse(veg, Imafter55);
MSE_45 = mse(veg, Imafter45);
MSE_44 = mse(veg, Imafter44);
MSE_34 = mse(veg, Imafter34);
MSE_33 = mse(veg, Imafter33);
MSE_22 = mse(veg, Imafter22);
MSE_13 = mse(veg, Imafter13);
MSE_12 = mse(veg, Imafter12);

MSEtot = [MSE_77 MSE_55 MSE_45 MSE_44 MSE_34 MSE_33 MSE_22 MSE_13 MSE_12]; %total
MSE
CatName = categorical({'7X7', '5X5', '4X5', '4X4', '3X4', '3X3', '2X2', '1X3', '1X2'});
%names of filters sizes

figure %plotting the MSE
bar(CatName, MSEtot)
title('MSE after filtering with different filters sizes')
ylabel('MSE')
grid on

figure %best filter and good image
subplot(1,2,1)
imshow(veg)
title('original clean image')
subplot(1,2,2)
imshow(Imafter33)
title('image after using median filter 3X3')

%% 2.2

```



```

clear all

veg = imread('vegetables.tif'); %loading image
veg = im2double(veg);

MSE_77 = zeros(1,100); %MSE intializing
MSE_55 = zeros(1,100);
MSE_45 = zeros(1,100);
MSE_44 = zeros(1,100);
MSE_34 = zeros(1,100);
MSE_33 = zeros(1,100);
MSE_22 = zeros(1,100);
MSE_13 = zeros(1,100);
MSE_12 = zeros(1,100);

m = [7 5 4 4 3 3 2 1 1]; %size of filter
n = [7 5 5 4 4 3 2 3 2];

for i = 1:100 %repeting exp 100 times
    NoiseVeg = imnoise(veg,'salt & pepper',0.2); %aadding noise

    Imafter77 = CleanSP(NoiseVeg,'Median',m(1),n(1)); %filtering the image using
    median filter with different sizes
    Imafter55 = CleanSP(NoiseVeg,'Median',m(2),n(2));
    Imafter45 = CleanSP(NoiseVeg,'Median',m(3),n(3));
    Imafter44 = CleanSP(NoiseVeg,'Median',m(4),n(4));
    Imafter34 = CleanSP(NoiseVeg,'Median',m(5),n(5));
    Imafter33 = CleanSP(NoiseVeg,'Median',m(6),n(6));
    Imafter22 = CleanSP(NoiseVeg,'Median',m(7),n(7));
    Imafter13 = CleanSP(NoiseVeg,'Median',m(8),n(8));
    Imafter12 = CleanSP(NoiseVeg,'Median',m(9),n(9));

    MSE_77(i) = mse(veg,Imafter77); %MSE for each filtering
    MSE_55(i) = mse(veg,Imafter55);
    MSE_45(i) = mse(veg,Imafter45);
    MSE_44(i) = mse(veg,Imafter44);
    MSE_34(i) = mse(veg,Imafter34);
    MSE_33(i) = mse(veg,Imafter33);
    MSE_22(i) = mse(veg,Imafter22);
    MSE_13(i) = mse(veg,Imafter13);
    MSE_12(i) = mse(veg,Imafter12);

end

MEAN_77 = mean(MSE_77); %mean of MSE
MEAN_55 = mean(MSE_55);
MEAN_45 = mean(MSE_45);
MEAN_44 = mean(MSE_44);
MEAN_34 = mean(MSE_34);
MEAN_33 = mean(MSE_33);
MEAN_22 = mean(MSE_22);
MEAN_13 = mean(MSE_13);
MEAN_12 = mean(MSE_12);

STD_77 = std(MSE_77); %std of MSE
STD_55 = std(MSE_55);
STD_45 = std(MSE_45);
STD_44 = std(MSE_44);
STD_34 = std(MSE_34);

```

```

STD_33 = std(MSE_33);
STD_22 = std(MSE_22);
STD_13 = std(MSE_13);
STD_12 = std(MSE_12);

meanMSEtot = [MEAN_77 MEAN_55 MEAN_45 MEAN_44 MEAN_34 MEAN_33 MEAN_22 MEAN_13
MEAN_12]; %total MEAN MSE
stdMSEtot = [STD_77 STD_55 STD_45 STD_44 STD_34 STD_33 STD_22 STD_13 STD_12];
%total STD MSE

CatName = categorical({'7X7','5X5','4X5','4X4','3X4','3X3','2X2','1X3','1X2'});
%names of filters sizes

figure %plotting the MSE
sgtitle('MSE after filtering with different filters sizes for 100 times')
subplot(1,2,1)
bar(CatName,meanMSEtot)
title('mean')
ylabel('MSE')
grid on
subplot(1,2,2)
bar(CatName,stdMSEtot)
title('std')
ylabel('MSE')
grid on

MSEtable = table(MSE_77', MSE_55', MSE_45' ,MSE_44', MSE_34' ,MSE_33' ,MSE_22'
,MSE_13',MSE_12'); %creating a table

MSEtable.Properties.VariableNames{1} = '7X7';
MSEtable.Properties.VariableNames{2} = '5X5';
MSEtable.Properties.VariableNames{3} = '4X5';
MSEtable.Properties.VariableNames{4} = '4X4';
MSEtable.Properties.VariableNames{5} = '3X4';
MSEtable.Properties.VariableNames{6} = '3X3';
MSEtable.Properties.VariableNames{7} = '2X2';
MSEtable.Properties.VariableNames{8} = '1X3';
MSEtable.Properties.VariableNames{9} = '1X2';

MSEtot = [MSE_77'; MSE_55'; MSE_45' ;MSE_44'; MSE_34' ;MSE_33'; MSE_22' ;MSE_13'
;MSE_12']; %total MSE

MSE_total = readtable('MSEtotal.txt'); %MSE for boxplot
% boxplot(MSE_total{:,1},MSE_total{:,2})

%using anova test
% p = anova(MSE_total{:,1},{MSE_total{:,2}});
[p,anovatab] = anova1(MSE_total{:,1},MSE_total{:,2});
title('boxplot')
xlabel('filter size')
ylabel('MSE')

%% Q3
clear all

%% loading the images
pics = load('exp1_pics.mat'); %loading the calibration images

```

```

number1 = pics.pics(1);
number2 = pics.pics(2);
number3 = pics.pics(3);
number4 = pics.pics(4);

number1 = number1{1,1};
number2 = number2{1,1};
number3 = number3{1,1};
number4 = number4{1,1};

%% finding Iris location for each eye
% q 3.1
cali1_3 = number1{3};
cali1_3 = rgb2gray(cali1_3); %color to BW photo
cali1_3 = im2double(cali1_3);
cali1_3 = cali1_3(230:277,155:390);

IMAGE = cali1_3; %current Frame
[Eye_Pos]= EyePosition_FUNC(IMAGE); %finding poistion of Irises for each eye in
one frame
N = length(IMAGE(1,:));

figure %plotting the result
sgtitle('Iris found using the function')
subplot(1,2,2)
imshow(IMAGE(:,ceil(N/2+10):end))
viscircles(Eye_Pos(2,:), 4, 'Color', 'b');
title('Right eye')
subplot(1,2,1)
imshow(IMAGE(:,1:ceil(N/2)))
viscircles(Eye_Pos(1,:), 4, 'Color', 'b');
title('Left eye')

%% creating Clibration frames matrix
% q 3.3
CalibFrames = [number1 ; number2 ; number3 ; number4]; %creating calibration frame
matrix
CalibFrames = CalibFrames(:,[1:2,4:end]);
CalibFrames = CalibFrames(:,[1:9,11:end]);
CalibFrames = CalibFrames(:,[1:2,4:end]);
CalibFrames = CalibFrames(:,[1:7,9:end]);
CalibFrames = CalibFrames(:,[1:5,7:end]);

eye_side = [1 2]; %1 for left eye, 2 for right eye
EyeCalib_L = EyeCalibration_FUNC(CalibFrames,eye_side(1)); %returns iris position
matrix
EyeCalib_R = EyeCalibration_FUNC(CalibFrames,eye_side(2)); %returns iris position
ma

figure %plotting histogram for each eye
histogram2(EyeCalib_L(1,:),EyeCalib_L(2,:),10)
title('Iris indx for left eye')
xlabel('x [pixel]')
ylabel('y [pixel]')
zlabel('counts')
view(-57,30)

figure
histogram2(EyeCalib_R(1,:),EyeCalib_R(2,:),10)

```

```

title('Iris indx for right eye')
xlabel('x [pixel]')
ylabel('y [pixel]')
zlabel('counts')

%% q 3.4

CalibFrames = CalibFrames(:,1:end-1); %taking onli N-1 from calibration, the last
one will be for checking the function functionality
FRAMES = CalibFrames(:,end);

%correcting the frames for each number
Frame1 = FRAMES{1,1};
Frame1 = rgb2gray(Frame1); %color to BW photo
Frame1 = im2double(Frame1);
Frame1 = Frame1(230:277,155:390);

Frame2 = FRAMES{2,1};
Frame2 = rgb2gray(Frame2); %color to BW photo
Frame2 = im2double(Frame2);
Frame2 = Frame2(230:277,155:390);

Frame3 = FRAMES{3,1};
Frame3 = rgb2gray(Frame3); %color to BW photo
Frame3 = im2double(Frame3);
Frame3 = Frame3(230:277,155:390);

Frame4 = FRAMES{4,1};
Frame4 = rgb2gray(Frame4); %color to BW photo
Frame4 = im2double(Frame4);
Frame4 = Frame4(230:277,155:390);

Eye_Look1 = EyeLook_FUNC(Frame1, EyeCalib_R); %returns the number with the highest
probability the subject looked at in the frame
Eye_Look2 = EyeLook_FUNC(Frame2, EyeCalib_R);
Eye_Look3 = EyeLook_FUNC(Frame3, EyeCalib_R);
Eye_Look4 = EyeLook_FUNC(Frame4, EyeCalib_R);

%% q 3.5

%segmentation by looking at the video
% 0:1 sec 1
% 1:3 sec 2
% 3:5 sec 3
% 5:7 sec 2
% 7:8 sec 3
% 8:10 sec 4
% 10:11 sec 3
% 11:13 sec 2
% 13:14 sec 1
% 14:16 sec 2
% 16:17 sec 1
% 17:18 sec 2
% 18:20 sec 3
% 20:22 sec 1
% 22:24 sec 2

```

```

numVecSegmentation = [1*ones(1,30) 2*ones(1,2*30) 3*ones(1,2*30) 2*ones(1,2*30)
3*ones(1,30) 4*ones(1,2*30) 3*ones(1,30)...
2*ones(1,2*30) 1*ones(1,30) 2*ones(1,2*30) 1*ones(1,30) 2*ones(1,30)
3*ones(1,2*30) 1*ones(1,2*30) 2*ones(1,2*30) 1*ones(1,2*30)];
Error = zeros(1,length(numVecSegmentation)); %error vector

videoEye = VideoReader('exp1video5.avi');
% vidEye = read(videoEye);
i = 1;
while hasFrame(videoEye)
    frameVID = readFrame(videoEye);
    frameVID = rgb2gray(frameVID); %color to BW photo
    frameVID = im2double(frameVID);
    frameVID = frameVID(230:277,155:390);

    Eye_Look = EyeLook_FUNC(frameVID, EyeCalib_R);

    if numVecSegmentation(i) == Eye_Look
        Error(i) = 0;
    else
        Error(i) = 1;
    end
    % Error(i)
    i = i + 1;
end

Error = Error(1:720); %without last frames

figure
stem(Error)
% set(gcf, 'Position', [100,500, 500, 400]);
title('Error for each frame','FontSize',18)
xlim([0 720])
ylim([0 1.1])
xlabel('frame number','FontSize',16)
ylabel('error','FontSize',16)

numberErrors = sum(Error); %number of errors

%% Q-4 -----
%% 4

clear all
clc
tic

v = VideoReader('exp2_video2.avi');
vid = read(v);

back = load("background.mat").pics{1,1};
background=double(rgb2gray(cell2mat(back(1,1))))/255;
flag = ones(1,v.NumFrames);

SE11 = strel('disk',2);%erosion the left side
SE22 = Negative(SE11.Neighborhood);%erosion the left side

for i = 1 : v.NumFrames %for loop to apply algorithm each time
    a=i

```

```

Video(i).Num = i;
    % saving each frame as BW
    Video(i).frames = rgb2gray(vid(:,:,:(i)));
    Video(i).frames = double(Video(i).frames)/255;
    Video(i).background = Video(i).frames-background; %subtracting the background
    Video(i).background1 = Video(i).background ;
    Video(i).background(Video(i).background>0.1) = 1; %high lighting the walking man
    Video(i).background(Video(i).background<-0.4) = 1; %high lighting the walking
man

    % Video(i).mask = Video(i).background;
    % a figure mask without the markers
    Video(i).mask_nomarker=imclose(Video(i).background,strel('disk',3));
    Video(i).mask_nomarker=imerode(Video(i).mask_nomarker,strel('disk',6));
    % mask eith good shape of the walking man
    %Video(i).mask_figure=imerode( Video(i).background,strel('disk',6));

% Video(i).mask = Video(i).mask_figure+2*Video(i).mask_nomarker; % the mask will
be combination of the two earlier masks
% Video(i).mask(Video(i).mask<=0.5)=0; % binary mask
% Video(i).mask(Video(i).mask>0.5)=1;

    Video(i).spots=imopen( Video(i).background,strel('disk',4)); %walking man with
figure complitly white with 4 black spots
% important
    Video(i).spots(Video(i).spots<=0.5)=0;
    Video(i).spots(Video(i).spots>0.5)=1;

    Video(i).lables=Video(i).spots+Negative(Video(i).mask_nomarker); %adding the
negative leaving us with only black spots

    Video(i).lables = imclose(Video(i).lables,strel('disk',4)); %removing noise

    temp_black_photo = zeros(360,640);
    temp_black_photo(120:330,170:360) = Negative(Video(i).lables(120:330,170:360));
    Video(i).location = temp_black_photo;
    %Video(i).location = imdilate(Video(i).location,strel('disk',4)) %removing noise
    Video(i).location(Video(i).location<=0.5)=0;

    Centroid
=cell2mat(struct2cell(regionprops(bwlabel(Video(i).location,4),'Centroid')));
    n=length(Centroid)/2;
    Centroid = reshape(Centroid,2,n)';
    Video(i).Coordinates = zeros(4,2);
    if n>4
        Video(i).Coordinates=Centroid(end-3:end,:);
        n=4;
    elseif n == 4
        Video(i).Coordinates = Centroid;
    else
        flag(1,i) = 0;
    end

    clear Centroid
end
%erosion----
Location = Video(logical(flag));

frNum = 13;

```

```

figure
subplot(231)
imshow(Locaition(frNum).frames)
title('Original frame','FontSize',18)
subplot(232)
imshow(Locaition(frNum).background1)
title('Original minus Background','FontSize',18)
subplot(233)
imshow(Locaition(frNum).background)
title('Background Thresholding ','FontSize',18)
subplot(236)
imshow(Locaition(frNum).mask_nomarker)
title('Mask without marker','FontSize',18)
subplot(235)
imshow(Negative(Locaition(frNum).spots))
title('Mask with marker','FontSize',18)
subplot(234)
imshow(Locaition(frNum).location)
title('4 markers','FontSize',18)

t =[Locaition.Num]/30 ;
coordinate_mat = [Locaition.Coordinates];
N = length(t);
%ploting space over time

x = zeros(4,N);
y = zeros(4,N);

for i = 1:N
[~,I] = sort(coordinate_mat(:,2*i));
coordinate_mat(1:4,(2*i-1):2*i) = coordinate_mat(I,(2*i-1):2*i);
x(1:4,i) = coordinate_mat(1:4,2*i-1);
y(1:4,i) = coordinate_mat(1:4,2*i)*(-1); % in the matrix the y axis is positivly
down
end

figure
plot3(x,t,y)
title('Locatin in space as function of time view (0,0)')
xlabel('x')
ylabel('time [s]')
zlabel('y')
view(0,0)
grid on
legend('Hip','Thigh','Knee','Ankle')

figure
plot3(x,t,y)
title('Locatin in space as function of time view (-35,30)')
xlabel('x')
ylabel('time [s]')
zlabel('y')
view(-35,30)
grid on
legend('Hip','Thigh','Knee','Ankle')

%4.2
distx = abs(diff(x')));

```

```

disty = abs(diff(y')));
distance = sqrt(distx.*distx+disty.*disty);
tdif=diff(t);
v = distance./diff(t);

figure
subplot(221)
plot (t(1:end-1),v(1,:))
title('Hip')
xlabel('time [s]')
ylabel('Velocity [pixel/s]')
subplot(222)
plot (t(1:end-1),v(2,:))
title('Thigh')
xlabel('time [s]')
ylabel('Velocity [pixel/s]')
subplot(223)
plot (t(1:end-1),v(3,:))
title('Knee')
xlabel('time [s]')
ylabel('Velocity [pixel/s]')
subplot(224)
plot (t(1:end-1),v(4,:))
title('Ankle')
xlabel('time [s]')
ylabel('Velocity [pixel/s]')

Walking_distance = (sum(v(4,1:end))/14)*10^(-3);

angle = atand(diff(x)./diff(y));

figure
subplot(121)
plot (t,angle(1,:))
title('Hip to Thigh')
xlabel('time [s]')
ylabel('Angle [degree]')
subplot(122)
plot (t,angle(2,:))
title('Thigh to Knee')
xlabel('time [s]')
ylabel('Angle [degree]')
toc

```