מעבדה בעיבוד אותות

פזיולוגיים

הנדסה ביורפואית

מגישים:

סול אמארה

דן טורצקי

תאריך:

07.11.2022

תוכן עניינים :

1 תשובות לשאלות הכנה:

שאלה 1:

1.1 כדי לייצג פיקסל אחד כאשר יש 256 רמות אפור דרושים $\log_2 256 = 8$ ביטים. בדומה לכך,

עבור תמונת RGB בה כל צבע מקבל 256 רמות דרושים : $3 \cdot \log_2 256 = 24$

כאשר יש 20 רמות אפור דרושים $\lceil \log_2 20 \rceil = 5$ ביטים לפיקסל. (ערך שלם כלפי מעלה כיוון

שעבור 4 ביטים נקבל רק 16 רמות וזה לא מספיק).

עבור תמונה צבעונית 1024x1024 עם 256 רמות דרושים $\log_2 256 \cdot 1024 \cdot 1024 = 2^{23}$

1.2

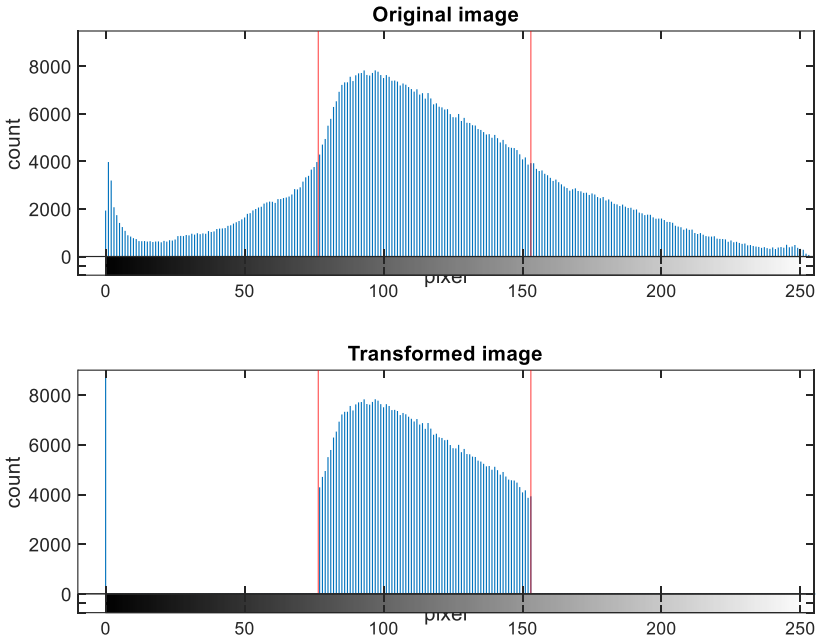
איור 1 : התמונה המקורית בגווני אפור

איור 2 : התמונה לאחר הטרנספורמציה

מאיור זה ניתן לראות את תוצאת הטרנפורמציה שביצענו המעבירה כל פיקסל שערכו גדול מ0.6

(בהירים) או קטן מ0.3 (כהים) ל0. ניתן להבחין כי פיקסלים אלה נצבעו בשחור המיוצג על ידי

הספרה 0 כרצוי. לדוגמא, עבור איזורי הפנים ברוב האנשים הצבע בהיר יותר ולכן הם נצבעו בשחור.



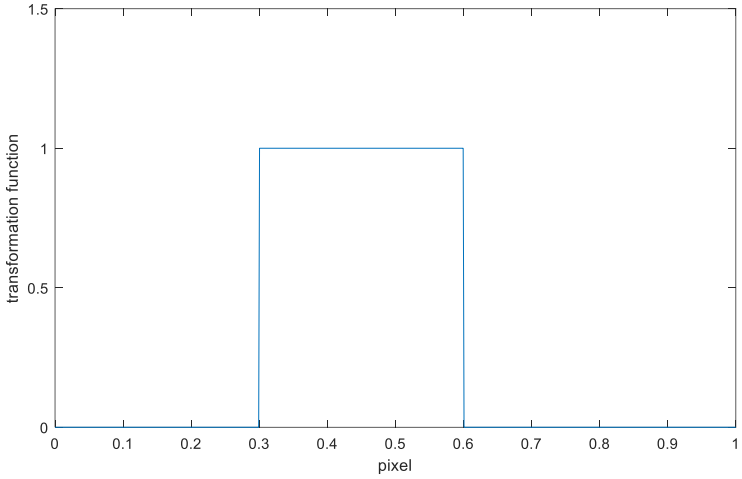
איור 3 : היסטוגרמת התמונות

מאיור זה ניתן לראות את ההיסטוגרמות של שתי התמונות כאשר באדום מסומנים קווי הגבול של הטרנספורמציות ($0.3 \cdot (2^8 - 1) = 76.5$, $0.6 \cdot (2^8 - 1) = 153$). נשים לב כי כיוון שהתחום מכיל את אפס, יש להחסיר 1 ולכן נכפיל ב255 ולא ב256. מאיורים אלה ניתן לראות כי עבור התמונה שלאחר הטרנספורמציה אין ערכים מעבר לתחום הרצוי מלבד הערכים באפס כנדרש.

המשוואה שמתארת את הטרנספורמציה הינה:

$$f(x) = \begin{cases} 0 & x \leq 0.3 \\ 1 & 0.3 < x < 0.6 \\ 0 & x > 0.6 \end{cases}$$

איור הטרנספורמציה:



איור 4 : פונקציית הטרנספורמציה

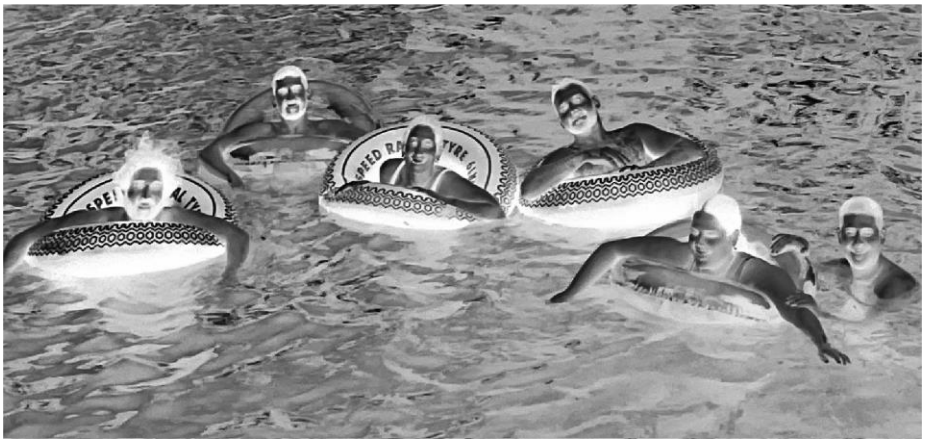
ניתן לראות כי פונקציית הטרנספורמציה הינה פונקציית מלבן המאפסת פיקסלים מחוץ לתחום [0.3 0.6].

1.3

פונקציית הנגטיב:

```
function [N_IMG]=Negative(IMG)
% Inputs:
% IMG - Gray levels image
%
% Outputs:
% N_IMG - Negative image of IMG
%%
R = 2^nextpow2(max(IMG(:))); %because IM is in image type variable, matlab
reduces 1 from the result automaticly
N_IMG = R-IMG;
```

1.4

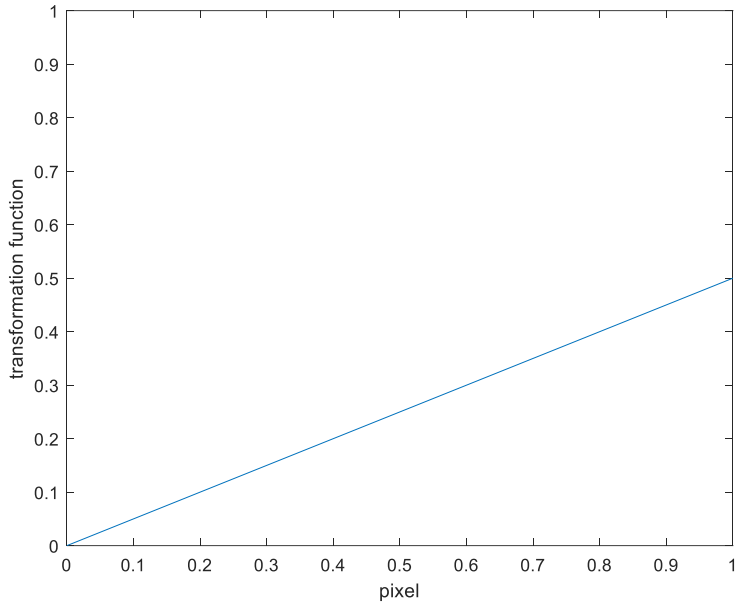


איור 5 : תמונת הנגטיב

פעולת הנגטיב מתמטית הינה אחד פחות ערכי הפיקסלים (כאשר 1 לבן, כלומר הערך המקסימלי שפיקסל יכול לקבל) ולכן היא הופכת את הבהירות של הפיקסלים – פיקסלים כהים לבהירים ולהיפך. בתמונה זו ניתן לראות כי האנשים שהיו בעלי פיקסלים יחסית בהירים הפכו להיות אפור כהה. דוגמה נוספת הינה גלגלי המים השחורים שהפכו להיות לבנים. מכאן נסיק כי הפונקציה שכתבנו מבצעת את הפעולה הנדרשת.

1.5

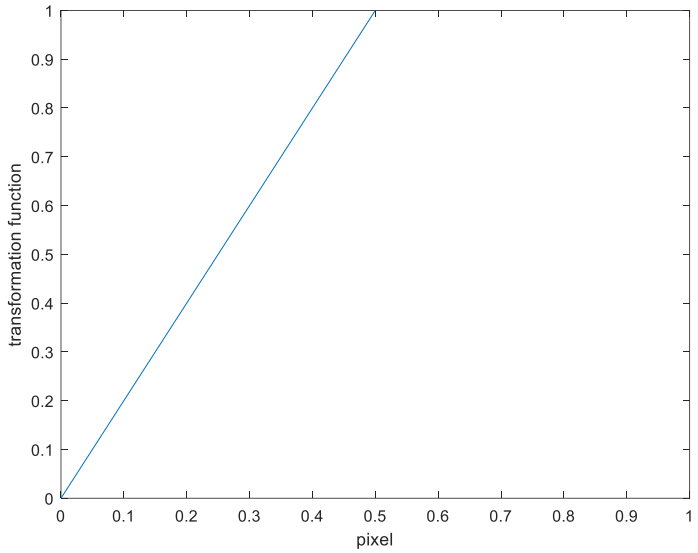
הטרנספורמציה הינה פונקציה הפועלת על כל איברי המטריצה של התמונה, לכן כדי שהטרנספורמציה תהיה הפוכה דרוש כי פונקציית הטרנספורמציה תהיה פונקציה הפיכה, כלומר חח״ע ועל. דוגמה לטרנספורמציה לינארית הפיכה הינה: $\mathrm{f(x)} = 0.5x$. הפונקציה ההפיכה לה הינה: $f^{-1}(x) = 2x$. איור הטרנספורמציה:



איור 6: טרנספורמציה לינארית הפיכה

ניתן לראות כי הערכים היוצאים נשמרים בטווח $[0, 1]$.

איור הטרנספורמציה ההפוכה:



איור 7 : הטרנספורמציה הלינארית ההפוכה

באיור זה, ניתן לראות כי עבור ערכי הפיקסלים בטווח $[0, 0.5]$ שהינו התמונה של הטרנספורמציה המקורית, תמונת הטרנספורמציה ההפיכה הינה $[0, 1]$.

נדגים את השימוש בטרנספורמציה הלינארית על התמונה מסעיף 1.2 :

איור 8 : התמונה מסעיף 1.2 לאחר הטרנספורמציה ההפיכה

כיוון שהטרנספורמציה מקטינה את ערכי כל הפיקסלים, התמונה נהייתה כהה יותר.

כעת, נשתמש בטרנספורמציה ההפוכה ונראה כי התמונה זהה לתמונה המקורית :

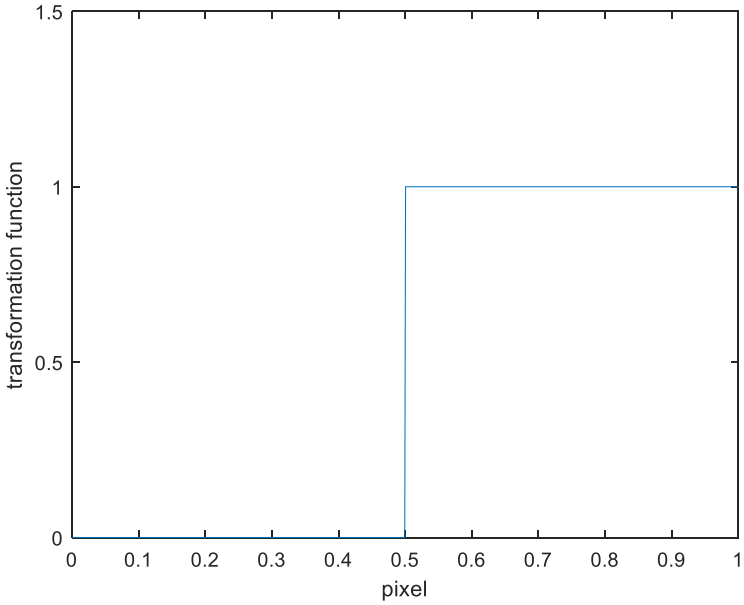
איור 9 : התמונה לאחר שימוש בטרנספורמציה ההפוכה

ניתן לראות כי תמונה זו זהה לתמונה המקורית מסעיף 1.2 ולכן הראנו כי הטרנספורמציה שביצענו הינה הפיכה.

דוגמא לטרנספורמציה לינארית למקוטעין שאינה הפיכה הינה :

$$f(x) = \begin{cases} 1 & x > 0.5 \\ 0 & else \end{cases}$$

איור הטרנספורמציה :



איור 10 : טרנספורמציה לינארית למקוטעין שאינה הפיכה

ניתן להבין כי טרנספורמציה זו אינה הפיכה, כיוון שערכים שלא ניתן לשחזר ערכים שאופסו.

1.6 Histogram equalization הינה שיטה שמטרתה לקחת תמונה עם טווח ערכי פיקסלים מצומצם שלא משתמש בכל הטווח ולהפוך אותו לטווח רחב יותר תוך שמירה על כמות הרמות ייצוג (L). בשל כך, פונקציית הטרנספורמציה צריכה לענות על שתי דרישות :

1. פונקציה מונוטונית עולה, על מנת לשמר את היחס בין הפיקסלים.

2. $0 \leq T(r) \leq L-1 \; for \; 0 \leq r \leq L-1$ על מנת לשמר את כמות הרמות.

הטרנספורמציה הנפוצה ביותר לשימוש הינה : $s = T(r) = (L-1)\int_0^r p_r(w)dw$ כאשר $p_r(w)$ הינה פונקציית צפיפות הסתברות. טרנספורמציה זו מעבירה את ההתפלגות של s להיות התפלגות אחידה. [1] פקודת המטלאב המתאימה הינה פונקציית histeq.

1.7 הגרסה הבדידה של הנוסחה הינה : $s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$

נסמן : $b_n = T(a_n)$ ונחשב :

$$b_n = T(a_n) = (L-1)\sum_{j=0}^{n} p_a(a_j)$$

$$T\big(T(a_n)\big) = T(b_n) = (L-1)\sum_{j=0}^{n} p_b(b_j) = (L-1)\cdot\sum_{j=0}^{n}\frac{1}{(L-1)} = n$$

כאשר המעבר אחד לפני האחרון נובע מכך שb מתפלג אחיד על פני הטווח $[0, L-1]$. תוצאה זו נובעת מכך שלאחר הטרנספורמציה הראשונה התפלגות הערכים הינה אחידה על פני כל הטווח ולכן אין צורך לבצע שינוי בערכים על מנת להגיע להתפלגות אחידה.

שאלה 2:

2.1 הבעיה נובעת מכך שבקצוות התמונה הסינון המרחבי לא כל איברי הפילטר משפיעים על התמונה (כיוון שזוהי קונבולציה דו מימדית וזה קורה בדומה לתופעת הקצוות בקונבולוציה חד מימדית).

פתרונות לבעיה-

1. חיתוך הקצוות
2. הורדת ערכי הקצוות על ידי הכפלה בגאוסיאין
3. ריפוד באפסים

לגבי שיטת חיתוך הקצוות, יתרון: שיטה פשוטה למימוש, חסרון: איבוד מידע של קצוות התמונה.

2.2 נוסחת הלפלסיאן הינה:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

ולכן

$$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

מכאן שהמטריצה המתארת את המסנן הינה:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

הפעלת המסנן על התמונה תבצע על ידי קונבולוציה של התמונה עם המטריצה שקיבלנו.

[1]

0	1	0	1	1	1
1	−4	1	1	−8	1
0	1	0	1	1	1
0	−1	0	−1	−1	−1
−1	4	−1	−1	8	−1
0	−1	0	−1	−1	−1

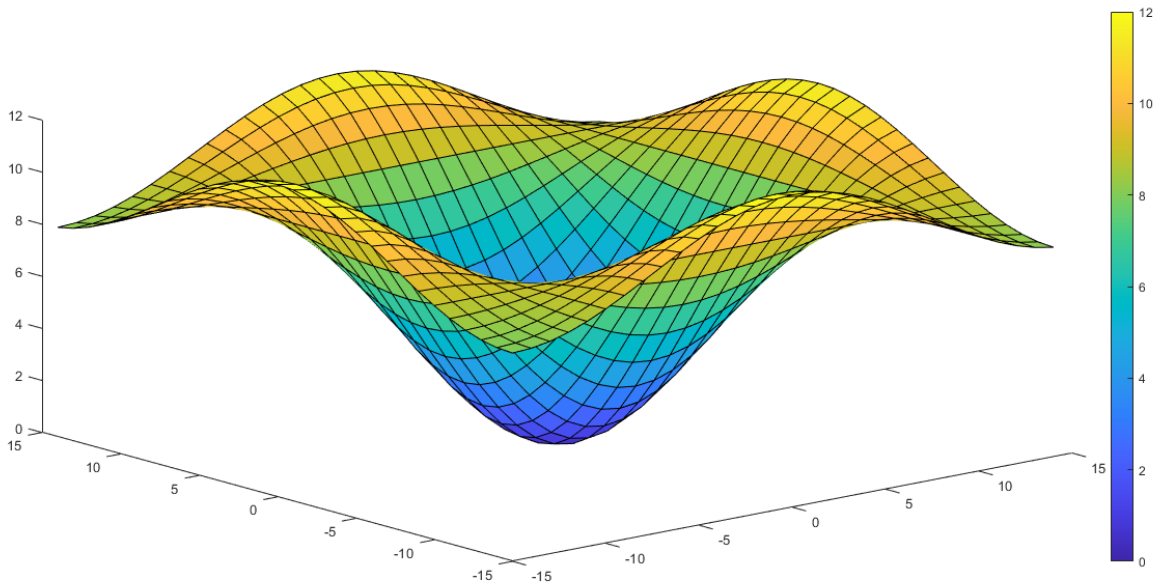
a b
c d

FIGURE 3.37 (a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

איור 11 : מסנן הלפלסיאן[1]

מאיור זה ניתן לראות את מסנני הלפלסיאן הפרקטיים בהם משתמשים תוך התחשבות בערכי האלכסונים (b,d).

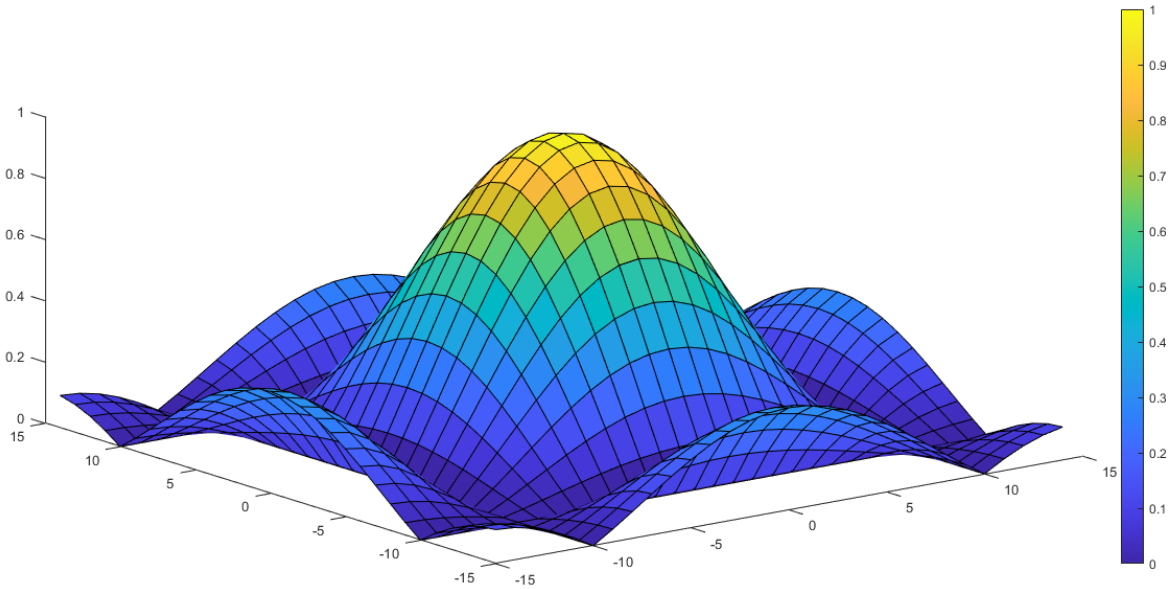
2.3



איור 12 : FFT עבור מסנן הלפלסיאן

מאיור זה ניתן לראות את תוצאת הFFT של מסנן הלפלסיאן. ניתן להבחין כי במרכז הערכים הינם אפס ואילו ככל שמתרחקים ממנו הערכים גדלים. מכאן, ככל שהתדרים גדלים נקבל ערכים

גבוהים יותר דבר התואם למסנן HPF. תוצאה זו הגיונית כיוון שלפלסיאן הינו נגזרת שנייה מרחבית ופעולת הנגזרת הינה פעולה של מסנן HPF.



איור 13 : FFT עבור מטריצת המיצוע

מאיור זה ניתן לראות את תוצאת הסינון עבור מטריצת המיצוע. ניתן להבחין כי במרכז הערכים הינם גבוהים ואילו ככל שמתרחקים ממנו הערכים קטנים ומתאפסים בקצוות. מכאן, ככל שהתדרים גדלים נקבל ערכים נמוכים יותר דבר התואם למסנן LPF. תוצאה זו הגיונית כיוון שמיצוע הינה דומה לאינטגרציה שמהווה LPF, כלומר מחליקים את המעברים בין הערכים השונים של פיקסלים שכנים.

על מנת לשפר תמונה מטושטשת נשתמש במסנן הראשון (לפלסיאן) כיוון שנרצה להדגיש הבדלים בין הפקסלים כלומר להשתמש במסנן מסוג HPF.

על מנת לטשטש תמונה נשתמש במסנן השני (מיצוע) כיוון שנרצה להוריד את החדות בין פיקסלים שכנים כלומר להשתמש במסנן מסוג LPF.

על מנת לזהות גבולות בתמונה, נרצה להדגיש את ההבדלים של הגבולות כלומר את השינויים החדים בן הצבעים ולכן נרצה להשתמש במסנן מסוג HPF – לפלסיאן. ניתן לבחור ערך סף מסוים

שפיקסלים עם ערכים גבוהים ממנו יוגדרו כערך של גבול כיוון שבגבולות נצפה לקבל ערכים גבוהים יותר (נגזרות).

2.4 פעולת הסינון הינה קונבולוציה בזמן של המסנן עם התמונה הרצויה. בשל כך, כאשר נרצה לעבור לתדר נצטרך להכפיל בין ההתמרות של המסנן לבין ההתמרה של התמונה. על מנת לבצע את ההכפלה איבר-איבר (לא כפל מטריצות אלה כפל איבר-איבר לפי מיקום), נצטרך ששתי המטריצות יהיו באותו הגודל. על מנת לפתור בעיה זו, ניתן לבצע ריפוד באפסים בתחום המרחב וכך נשמור על תכונות המסנן בתדר ובעצם נבצע לו מתיחה לגודל הרצוי.

2.5

איור 14 : התמונה המקורית ולאחר שלושת הפילטרים

מאיור זה ניתן לראות את תוצאת הסינון של כל אחד מהפילטרים. נבחין כי בעזרת שלושתם קיבלנו טשטוש לתמונה כצפוי. כמו כן, ככל שגודל הפילטר גדל כך גם השפעתו על התמונה ולכן קיבלנו את הטשטוש הגבוה ביותר עבור מסנן 5x5. תוצאה זו נובעת מכך שככל שגודל הפילטר גדל כל איטרציה בקונבולוציה בעלת השפעה גדולה יותר על המוצא.

איור 15 : שיפור התמונות בשתי הדרכים

באיורזה ניתן לראות את התמונה המקורית עליה אנחנו רוצים לבצע את השיפור (מאיור 14).

העמודה שמאלית מייצגת את תוצאת השיפור עם פילטר לפלסיאן כפי שהסברנו בסעיף 2.4

והעמודה הימנית מייצגת את תוצאת השיפור לאחר שימוש בפונקציית fspecial במטלאב. בשני

סוגי השיפורים, הסינון מניב את קווי הגבולות אותן חיברנו לתמונה מאיור 14 לקבלת התמונה

המשופרת. ניתן לראות כי באמצעות הzero padding קיבלנו שיפור בתמונה לעומת התמונה

המקורית ואילו הפונקציה של המטלאב לא שיפרה את התמונה. ראשית, פונקציהזו עושה פילטר

3x3 וראינו כי ככל שהגודל של הפילטר גדול יותר הסינון יותר אפקטיבי. בנוסף, המסנן הינו :

$$\begin{bmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{2}{3} & -3\frac{1}{3} & \frac{2}{3} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix}$$

ולכן ייתכן שמכך התוצאות שלו פחות טובות.

נחשב את שגיאת הMSE:

$$ZeroPadding_{3x3} = 4110926\ , ZeroPadding_{4x4} = 3860462\ ,$$

$$ZeroPadding_{5x5} = 3466924$$

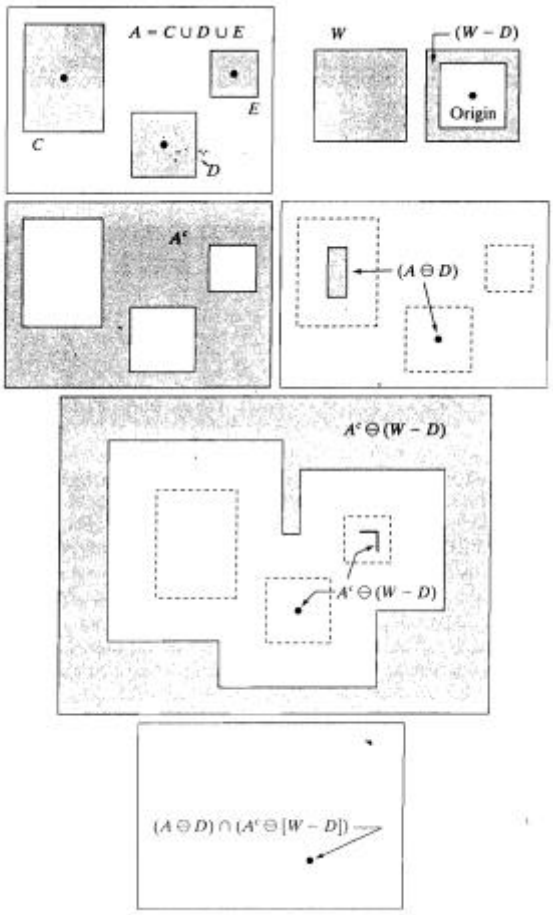
$$fspecial_{3x3} = 2445659, fspecial_{4x4} = 2607089, fspecial_{5x5} = 2540615$$

מכאן ניתן לראות כי למרות שתוצאות הסינון באמצעות *fspecial* למראית עין נראות פחות טובות, השגיאה שלהן מהתמונה המקורית קטנה יותר.

נסכם כי עדיין הדרך של ריפוד באפסים טובה יותר לדעתנו למרות שהשגיאה בה גדולה יותר מכיוון שהתמונה נראית פחות מטושטשת.

שאלה 3:

3.1



איור 16 : המחשת האלגוריתם

מטרת האלגוריתם למצוא את מיקום תבניות מסוימות בתוך תמונה. האלגוריתם מוצא את כל הנקודות של המרכזים בהן כל D (התבנית), מוכלת בA (איחוד של כל התבניות שקיימות בתמונה).

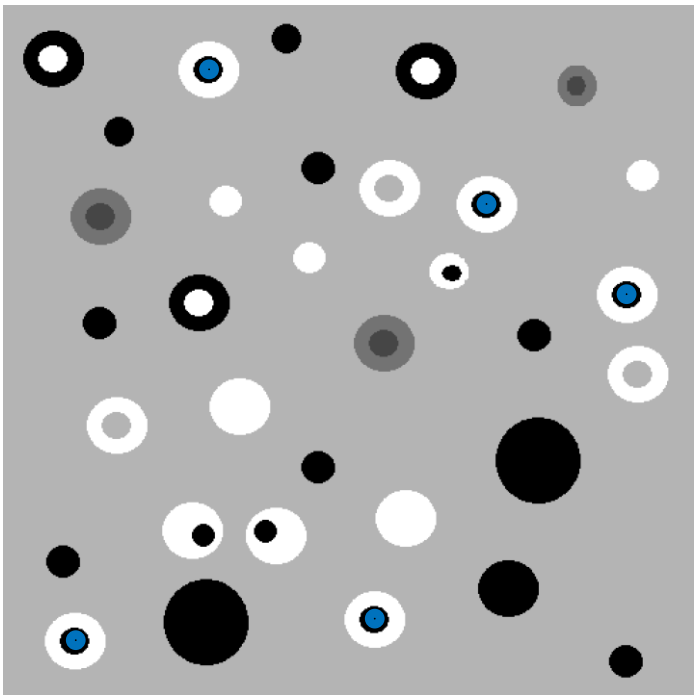
לאחר מכן האלגוריתם מוצא את כל המרכזים בהם (W-D) , המסגרת של D, מוכלת בתוך המשלים של A. לאחר מכן, החיתוך של שתי הקבוצות הנ"ל הינו המרכז של D המוכל בתוך A.

3.2 הפונקציה שכתבנו :

```
function [n,Coordinates]=LocateCirc(IMG)
% Inputs:
% IMG - Gray levels image
%
% Outputs:
% n - number of identified patterns.
% Coordinates - a vector with the XY coordinates of the
% required pattern [ n * 2 (X Y)]
```

```
figure
imshow(IMG);hold on;
xline(125,'color','r');xline(182,'color','r');yline(22,'color','r');
yline(75,'color','r')

shape=IMG(22:75,125:182);

maxIMG=max(double(max(max(IMG))));
modeIMG=max(double(mode(mode(IMG))));
label=modeIMG/maxIMG;
IMGbinary=im2bw(IMG,label-0.01);
shapeBINARY=im2bw(shape,label-0.01);

n=bwhitmiss(IMGbinary,shapeBINARY,1-shapeBINARY);
[x,y]=find(n==1);
Coordinates=[x,y];
n=length(x);
end
```

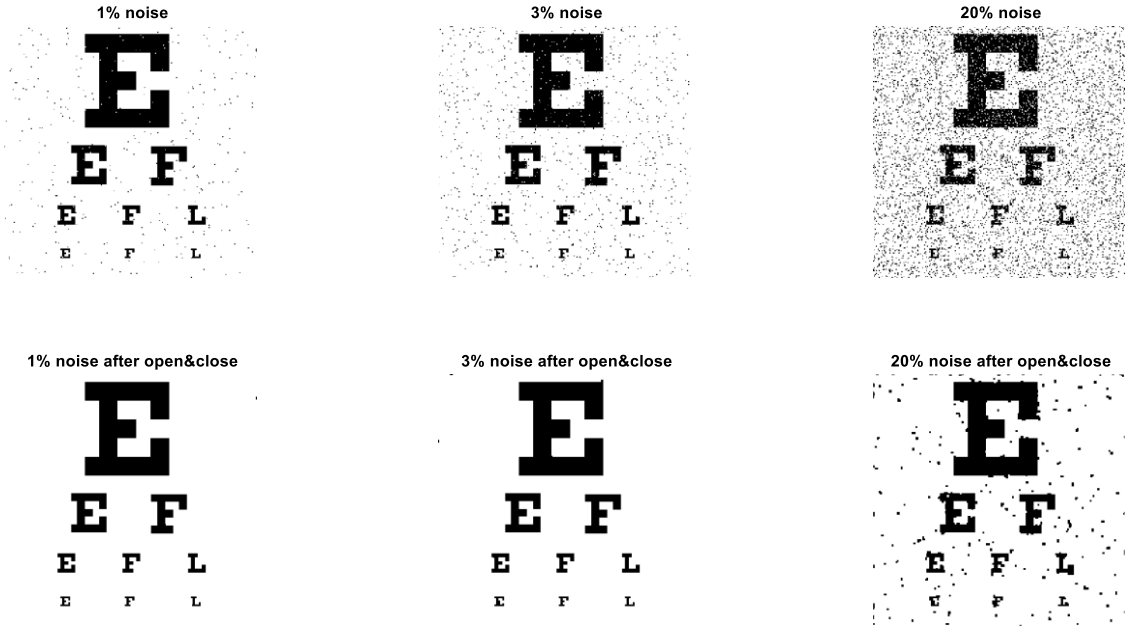
לאחר מציאת התבנית הרצויה (על ידי ניסוי וטעיה עם הקווים), העברנו את התמונות לבינאריות על ידי מציאת הערך השכיח ביותר בתמונה. כך קיבלנו את מרכזי העיגולים הרצויים בשחור ואת שאר התמונה בלבן. הפעלנו את פונקציית bwhitmiss במטלאב ומצאנו את מרכזי העיגולים. התוצאה:



איור 17: תוצאת האלגוריתם

מאיור זה ניתן לראות את תוצאת האלגוריתם. העיגולים הכחולים מסמנים את מרכזי העיגולים שזוהו וניתן לראות כי כולם זוהו נכון ואין טעויות כנדרש.

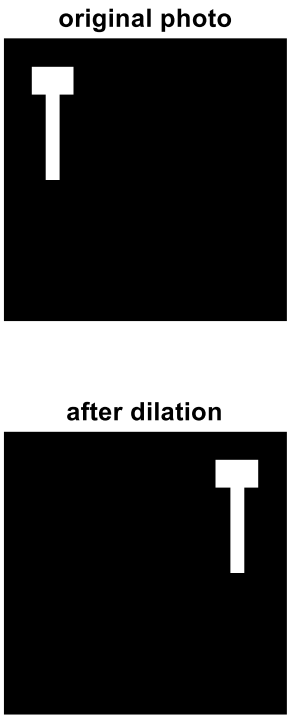
3.3



איור 18 : התמונות לאחר הוספת רעש וסינונו

מאיור זה ניתן לראות כי בשורה הראשונה קיבלנו את האות עם הרעש כרצוי. לאחר מכן בשורה השנייה רואים את הסינון שביצענו, כאשר עבור 1,3% קיבלנו סינון טוב שדומה לתמונה המקורית אבל עבור 20% לא הצלחנו לסנן את כל הרעש. רוב השגיאה נובעת מכך שלא הצלחנו להסיר את הרעש השחור בפקודת הopen אבל כן הצלחנו למלא את הרווחים באמצעות close.

3.4 כיוון האיש נמצא בצד שמאל ואנחנו רוצים להעביר אותו לימין, כלומר אנחנו רוצים לבצע הזזה על ציר x בלבד. בשל כך, בנינו מטריצה פשוטה : $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ שמטרתה להזיז את האיש כל פעם פיקסל אחד ימינה. חזרנו על פעולה זו בלולאה והתוצאה שהתקבלה הינה :



איור 19 : תוצאת הזזה האדם

ניתן לראות כי קיבלנו את התוצאה הרצויה והשינוי היחיד שנעשה הינו הזזה של האדם לחלק הימני של התמונה.

3.5 הפונקציה שכתבנו :

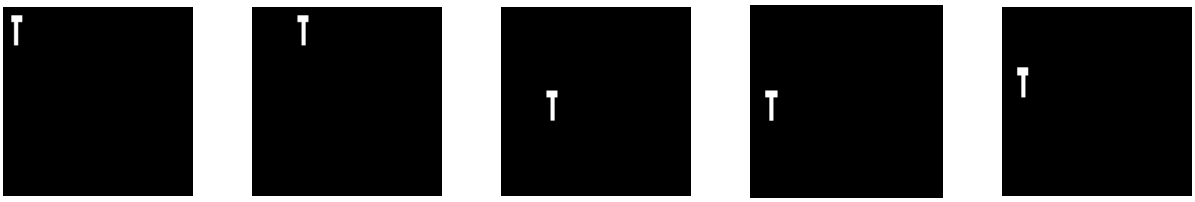
```
function [IMG1]=IMove(Track)
% Input:
% Track - a vector [n x 2] contains movement instruction for
% stick-man. the first column contains axis:
% 1=movement in the X-axis
% 2=movement in the Y-axis
% the second column contains number of steps in the same
% direction.
%
% Output:
% M - a Matrix [ 50 x 50 x (n*steps) ] contains the movements
% step-by-step.
IMG=zeros(50,50);
IMG(5:10,4)=1;
IMG(3:4,3:5)=1;

xR=[0,0,0;0,0,1;0,0,0]; %x-right
xL=[0,0,0;1,0,0;0,0,0]; %x-left
yU=[0,1,0;0,0,0;0,0,0];%y-up
yD=[0,0,0;0,0,0;0,1,0];%y-down
IMG1=IMG;
figure;subplot(length(Track(:,1))+1,1,1);imshow(IMG1);
for i=1:length(Track(:,1))
    stepdiraction=Track(i,1);
    num=Track(i,2);
```

```
    %decide the diraction:
    if stepdiraction==1
        if num>0
            mat=xR;
        else
            mat=xL;
        end
    else
        if num>0
            mat=yU;
        else
            mat=yD;
        end
    end

    for j=1:abs(num) %number of steps
    IMG1=imdilate(IMG1,mat);
    end

subplot(length(Track(:,1))+1,1,i+1);imshow(IMG1);
end

end
```

הדגמה עבור המטריצה:

$$\begin{bmatrix} 1 & 10 \\ 2 & -20 \\ 1 & -8 \\ 2 & 6 \end{bmatrix}$$

התוצאה שהתקבלה:



איור 20: שלבי התזוזה

האיור השמאלי ביותר מציג את התמונה המקורי. לאחר מכן רואים תזוזה בציר האיקס של 10 פיקסלים כרצוי. התמונה האמצעית מציגה את התזוזה המופיעה בשורה ה-2: 20 צעדים למטה (בכיוון y). וכך הלאה. ניתן לראות כי האדם יכול לזוז בכל ארבעת הכיוונים והתמונות מתקבלות כפי שרצינו.

3.6

א. האלמנט שיבצע את הפעולה הינו :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

.

פעולת הdilation עושה שיקוף לפי ציר x וy למטריצת האלמנט ואז בודקת אם יש חפיפה של לפחות איבר אחד בין המטריצה המשוקפת לתמונה. כל פעם שמים את המרכז על פיקסל אחר, ובמידה ויש חפיפה אחת לפחות הפיקסל מקבל 1.

ב. על מנת להרחיב ריבוע *2x2* לריבוע *4x4* נרצה להרחיב את הריבוע הלבן לכל אחד מהכיוונים ולכן האלמנט שיבצע פעולה זו הינו :

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2 מקורות

[1] R. C. Gonzalez, Digital image processing, 3rd ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.

3 נספחים

Main

```
IM=imread('IMG_4314.jpg');
IM = im2gray(IM);
figure; imshow(IM)

%1.2 image transformation
minb=0.3*(2^8- 1);
maxb=0.6*(2^8-1);
index=find(IM>maxb|IM<minb);
newIM=IM; newIM(index)=0;
figure; imshow(newIM)

%histograms
figure; subplot(2,1,1); imhist(IM); hold on; xline(maxb,'color','r');
xlim([-10 255]); xline(minb,'color','r'); xlabel('pixel');
ylabel('count');title('Original image')
subplot(2,1,2); imhist(newIM);
xline(maxb,'color','r');xline(minb,'color','r');ylim([0 9000]); xlim([-10
255]); xlabel('pixel'); ylabel('count');title('Transformed image')

%transofrmation function
x = linspace(0,1,800);
transform = ones(1,length(x));
transform(x<0.3 | x>0.6) = 0;
figure; plot(x,transform); xlabel('pixel'); ylabel('transformation
function'); ylim([0 1.5])

%1.4 negative
figure; imshow(Negative(IM))

%1.5
figure; plot(x,0.5*x); xlabel('pixel'); ylabel('transformation function');
ylim([0 1])
x2 = linspace(0,0.5,900);
figure; plot(x2,2*x2); xlabel('pixel'); ylabel('transformation function');
xlim([0 1])

tranformed_im = 0.5*IM;
figure;imshow(tranformed_im);
figure;imshow(2*tranformed_im);

%unreversable transformation
transform_2 = x>0.5;
figure; plot(x,transform_2); xlabel('pixel'); ylabel('transformation
function'); ylim([0 1.5])

%% 2:
%2.3
m1=[-1,-1,-1;-1,8,-1;-1,-1,-1];
m2=(1/9)*[1,1,1;1,1,1;1,1,1];

M1=fft2(m1,30,30);
M2=fft2(m2,30,30);
```

```
figure
hold on
surface(-15:14,-15:14,abs(fftshift(M1)));
colorbar();
view(3)

figure
hold on
surface(-15:14,-15:14,abs(fftshift(M2)));
colorbar();
view(3)

%2.5
lenna=imread('lenna.jpg');
m3=(1/9)*[1,1,1;1,1,1;1,1,1];
m4=(1/16)*[1,1,1,1;1,1,1,1;1,1,1,1;1,1,1,1];
m5=(1/25)*[1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1];
figure; imshow(lenna)
lenna3=imfilter(lenna,m3);
lenna4=imfilter(lenna,m4);
lenna5=imfilter(lenna,m5);

figure; subplot(2,2,1);imshow(lenna);title('original');
subplot(2,2,2);imshow(lenna3);title(' 3X3 Average filtering');
subplot(2,2,3);imshow(lenna4);title('4X4 Average filtering');
subplot(2,2,4);imshow(lenna5);title('5X5 Average filtering');

M1=fft2(m1,256,256);
L3=fft2(lenna3).*M1; lenna31=ifft2(L3);
L4=fft2(lenna4).*M1; lenna41=ifft2(L4);
L5=fft2(lenna5).*M1; lenna51=ifft2(L5);

fsp=fspecial('laplacian');


figure
subplot(3,3,1);imshow(uint8(lenna31)+lenna3); title('zero padding to 3X3');
subplot(3,3,2);imshow(lenna3); title('3X3 Average filtering');
subplot(3,3,3);imshow(imfilter(lenna3,fsp)+lenna3); title('fspecial to
3X3');
subplot(3,3,4);imshow(uint8(lenna41)+lenna4); title('zero padding to 4X4');
subplot(3,3,5);imshow(lenna4); title('4X4 Average filtering');
subplot(3,3,6);imshow(imfilter(lenna4,fsp)+lenna4); title('fspecial to
4X4');
subplot(3,3,7);imshow(uint8(lenna51)+lenna5); title('zero padding to 5X5');
subplot(3,3,8);imshow(lenna5); title('5X5 Average filtering');
subplot(3,3,9);imshow(imfilter(lenna5,fsp)+lenna5); title('fspecial to
5x5');

%MSE:
zp3=sum(sum((uint8(lenna31)+lenna3-lenna).^2))
zp4=sum(sum((uint8(lenna41)+lenna4-lenna).^2))
zp5=sum(sum((uint8(lenna51)+lenna5-lenna).^2))

fsp3=sum(sum((imfilter(lenna3,fsp)+lenna3-lenna).^2))
fsp4=sum(sum((imfilter(lenna4,fsp)+lenna4-lenna).^2))
fsp5=sum(sum((imfilter(lenna5,fsp)+lenna5-lenna).^2))
```

```
%% 3:
%3.2
IMG=imread("circles.png");
[n,Coordinates]=LocateCirc(IMG);
figure
imshow(IMG);hold on;
scatter(Coordinates(:,2),Coordinates(:,1),'LineWidth',5);

%3.3
IMG=imread('Iexam.tif');
IMGwithnoise1=imnoise(IMG,'salt & pepper',0.01);
IMGwithnoise3=imnoise(IMG,'salt & pepper',0.03);
IMGwithnoise20=imnoise(IMG,'salt & pepper',0.2);

SE = strel('cube',2);
I1=imopen(IMGwithnoise1,SE); I1=imclose(I1,SE);
I3=imopen(IMGwithnoise3,SE); I3=imclose(I3,SE);
I20=imopen(IMGwithnoise20,SE); I20=imclose(I20,SE);

figure;
subplot(2,3,1); imshow(IMGwithnoise1); title(' 1% noise')
subplot(2,3,2); imshow(IMGwithnoise3); title(' 3% noise')
subplot(2,3,3); imshow(IMGwithnoise20); title(' 20% noise')
subplot(2,3,4); imshow(I1); title(' 1% noise after open&close')
subplot(2,3,5); imshow(I3); title(' 3% noise after open&close')
subplot(2,3,6); imshow(I20); title(' 20% noise after open&close')

%3.4
IMG=zeros(20,20);
IMG(5:10,4)=1;
IMG(3:4,3:5)=1;
SE=[0,0,0;0,0,1;0,0,0];
IMG1=IMG;
for i=1:13
    IMG1=imdilate(IMG1,SE);
end

figure
subplot(2,1,1);imshow(IMG);title('original photo')
subplot(2,1,2);imshow(IMG1);title('after dilation')

%3.5
Track=[1,10;2,-20;1,-8;2,6];
IMG1=IMove(Track);
```

Negative

```
function [N_IMG]=Negative(IMG)
% Inputs:
% IMG - Gray levels image
%
% Outputs:
% N_IMG - Negative image of IMG
%%
R = 2^nextpow2(max(IMG(:))); %because IM is in image type variable, matlab
reduces 1 from the result automaticly
N_IMG = R-IMG;
```

LocateCirc

```
function [n,Coordinates]=LocateCirc(IMG)
% Inputs:
% IMG - Gray levels image
%
% Outputs:
% n - number of identified patterns.
% Coordinates - a vector with the XY coordinates of the
% required pattern [ n * 2 (X Y)]

figure
imshow(IMG);hold on;
xline(125,'color','r');xline(182,'color','r');yline(22,'color','r');yline(7
5,'color','r')

shape=IMG(22:75,125:182);

maxIMG=max(double(max(max(IMG))));
modeIMG=max(double(mode(mode(IMG))));
label=modeIMG/maxIMG;
IMGbinary=im2bw(IMG,label-0.01);
shapeBINARY=im2bw(shape,label-0.01);

n=bwhitmiss(IMGbinary,shapeBINARY,1-shapeBINARY);
[x,y]=find(n==1);
Coordinates=[x,y];
n=length(x);
end
```

imove

```
function [IMG1]=IMove(Track)
% Input:
% Track - a vector [n x 2] contains movement instruction for
% stick-man. the first column contains axis:
% 1=movement in the X-axis
% 2=movement in the Y-axis
% the second column contains number of steps in the same
% direction.
%
% Output:
% M - a Matrix [ 50 x 50 x (n*steps) ] contains the movements
% step-by-step.
IMG=zeros(50,50);
IMG(5:10,4)=1;
IMG(3:4,3:5)=1;

xR=[0,0,0;0,0,1;0,0,0]; %x-right
xL=[0,0,0;1,0,0;0,0,0]; %x-left
yU=[0,1,0;0,0,0;0,0,0];%y-up
yD=[0,0,0;0,0,0;0,1,0];%y-down
IMG1=IMG;
figure;subplot(1,length(Track(:,1))+1,1);imshow(IMG1);
for i=1:length(Track(:,1))
```

```
    stepdiraction=Track(i,1);
    num=Track(i,2);
    %decide the diraction:
    if stepdiraction==1
        if num>0
            mat=xR;
        else
            mat=xL;
        end
    else
        if num>0
            mat=yU;
        else
            mat=yD;
        end
    end

    for j=1:abs(num) %number of steps
    IMG1=imdilate(IMG1,mat);
    end

subplot(1,length(Track(:,1))+1,i+1);imshow(IMG1);
end

end
```