

```

function [Ix, Iy] = ImageDerivatives(im)
    % x derivation
    Kx = [0.5 -0.5];
    Ix = conv2(im, Kx, 'full');
    Ix = Ix(:, 1:end-1);
    Ix = BordersToZero(Ix);
    %show(Ix)

    % y derivation
    Ky = [-0.5; 0.5];
    Iy = conv2(im, Ky, 'full');
    Iy = Iy(1:end-1, :);
    Iy = BordersToZero(Iy);
    %show(Iy)

end

function [im] = BordersToZero(im)
    im(:, [1 end]) = 0;
    im([1 end], :) = 0;
end

function [L] = Deriv2Laplace(Ix, Iy)
    Kx = [0.5 -0.5];
    Ky = [-0.5; 0.5];
    L = conv2(Ix, Kx, 'same') + conv2(Iy, Ky, 'same');
end

function [R, L] = do_retinex(I,T)

    % Take log and calculate the derivatives of the log image.
    [Ix, Iy] = ImageDerivatives(log(I));

    % Calculate the norm of the derivative at each point
    norm = sqrt(Ix.^2 + Iy.^2);

    % Set all derivatives whose norm is less than T to zero.
    Ix(norm < T) = 0;
    Iy(norm < T) = 0;

    % Use the function Deriv2Laplace to compute the Laplacian.
    Laplas = Deriv2Laplace(Ix, Iy);

    % Compute the inverse Laplacian kernel K using the provided invDel2.
    [invK] = invDel2(size(I));

    % Convolve the Laplacian with K to obtain the log reflectance
    logRef = conv2(Laplas, invK, 'same');

    % Exponentiate the log reflectance to obtain a reflectance image.
    R = exp(logRef);
    L = I./R;

end

function q3
    im = double(imread('simul_cont_squares.tif'));
    [Ix, Iy] = ImageDerivatives(im);
    [L] = Deriv2Laplace(Ix, Iy);
    imshow(L);

    % after threshold
    threshold = 10;
    L = abs(L);
    L_threshold = L >= threshold;
    imshow(L_threshold);

end

function q4
    im = double(imread('cross.tif'));

```

```

[Ix, Iy] = ImageDerivatives(im);
[L] = Deriv2Laplace(Ix, Iy);
imshow(L);

% after threshold
threshold = 5;
L = abs(L);
L_threshold = L >= threshold;
imshow(L_threshold);

end

function q5
    im = double(imread('kofka_ring.tif'));
    [Ix, Iy] = ImageDerivatives(im);
    [L] = Deriv2Laplace(Ix, Iy);
    %imshow(L);

    % after threshold
    threshold = 20;
    L = abs(L);
    L_threshold = L >= threshold;
    imshow(L_threshold);

end

function q8

    [im] = twoSquares(1);
    show(im,[0 2])

    T = 0.07;
    [R1, L1] = do_retinex(im,T);
    d1 = diag(R1);
    imshow(R1)
    imshow(L1)
    plot(d1)

    [im] = twoSquares(2);
    show(im,[0 2])

    T = 0.07;
    [R2, L2] = do_retinex(im,T);
    d2 = diag(R2);
    plot(d2)
    imshow(R2)
    imshow(L2)

end

function q9

    load('checkerShadow.mat');
    figure;
    imshow(im1,[0 1])
    title('org image')

    % Verify that indeed the two checks have exactly the same intensity
    disp('Intensities at (y1,x1), (y2,x2):');
    disp([im1(y1,x1) im1(y2,x2)])

    T_lst = [0.02, 0.07, 0.2];
    for i = 1:length(T_lst)
        T = T_lst(i);
        %T = 0.07;
        [R, L] = do_retinex(im1,T);
        disp('calculated reflectance of the two checks, T is:');
        disp(T)
        disp([R(y1,x1) R(y2,x2)])
        figure;
        imshow(R)
        title_str = sprintf('R, T= %.2f', T);
        title(title_str);
    end

```

```

end

end

function ql0

    load('runner.mat');
    figure;
    imshow(iml, [0 1])
    title('org image')

    figure;
    T = 0.05:0.02:0.15;
    for i = 1:6
        [R, L] = do_retinex(iml, T(i));
        subplot(2,3,i);
        str = sprintf('R, T=%.2f', T(i));
        imshow(R, [0 2]);
        title(str);
    end

end

function ql1

    load('couch.mat');
    figure;
    imshow(iml, [0 1])
    title('org image')

    figure;
    T = 0.01:0.01:0.04;
    for i = 1:4
        [R, L] = do_retinex(iml, T(i));
        subplot(2,2,i);
        str = sprintf('R, T=%.2f', T(i));
        imshow(R, [0 1]);
        title(str);
    end

end

```

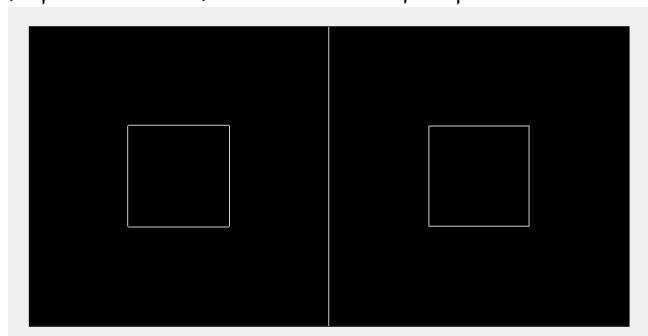
ראיה אנושית גישה חישובית | תרגיל 3 (12967)

שם: רונאל חרדים | ת"ז: 208917641

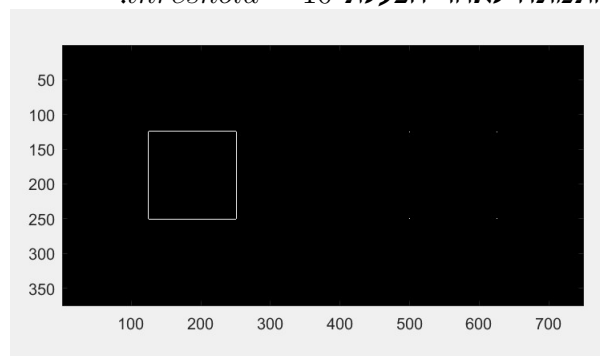
שאלה 3

הריבוע השמאלי שנמצא על הרקע הכהה, נראה לנו בהיר יותר.

תמונת הלפליסיאן שקיבלנו לאחר הפעלת הלפליסיאן על התמונה ללא $threshold$:

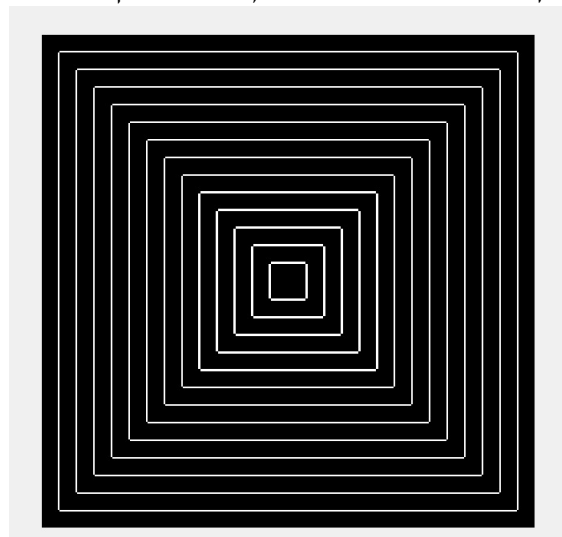


התמונה לאחר הפעלת $threshold = 10$:

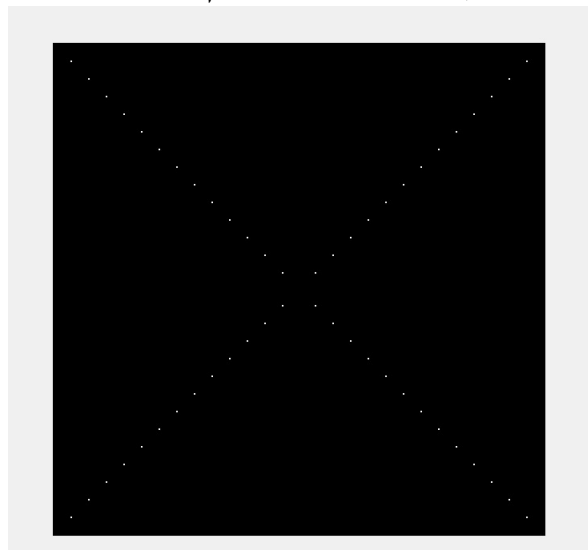


שאלה 4

לקחנו את התמונה *cross*, וכפי שניתן לראות אנחנו רואים את ה X על התמונה.

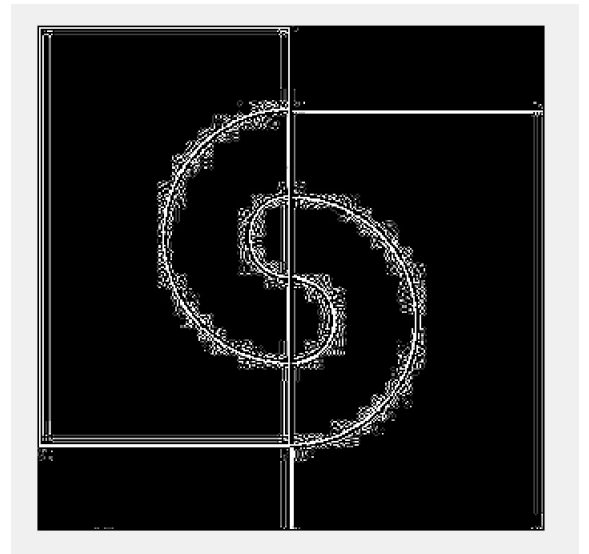


לאחר הפעלת $threshold = 5$ קיבלנו את התמונה הבאה:

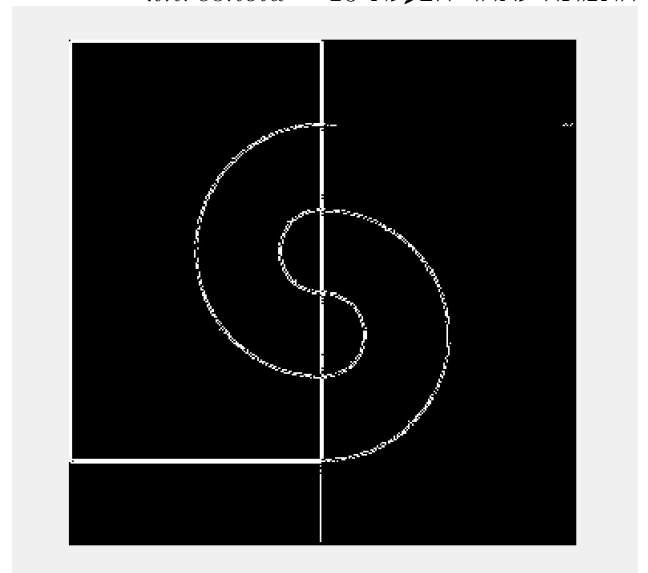


שאלה 5

התמונה לאחר הפעלת לפליסיאן:



התמונה לאחר הפעלת $threshold = 10$:



התמונה לאחר הפעלת $threshold = 20$:



הסבר:

ההסבר לאשליה הזאת מורכבת יותר, משום שאין גבול ברור של מעבר צבעים, אלא הטבעת מתפתלת על פני רקעים שונים. לכן כדי להסביר את התופעה הזאת נצטרך הסבר משוכלל יותר.

שאלה 6

את התופעה הראשונה אכן ניתן להסביר באמצעות לפלסיאן והבדלי צבע מהרקע. משום שהריבוע שנמצא על הרקע הבהיר יותר נראה לנו כהה יותר, ולהיפך. קל לראות את הבדלי הצבע.

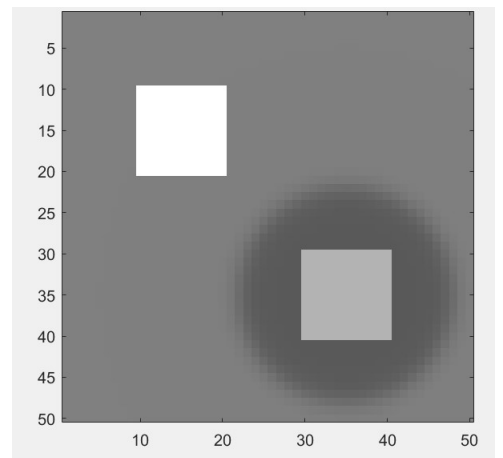
התמונה השניה, גם בה יש הבדלי צבע באופן מדורג, וניתן להסביר כי הבדלי הצבע יוצרים אצלנו נגזרת וכך אנו מבחינים בהבדל, וזה יותר את X .

את התמונה השלישית קשה יותר להסביר בשיטה זו, משום שהגבולות של שני חצאי הטבעת דועכים באותו קצב ביחד עם הגדלת הסף. ניתן להסביר אותה בעזרת אלגוריתמים שראינו בכיה, לפיהם העין האנושית מבצעת חישוב מורכב יותר, החל מהפיקסל הבהיר ביותר בתמונה שאותו היא משווה ללבן. לאחר מכן היא מבצעת חישובים למציאת $\rho(x, y)$ של כל פיקסל בתמונה, עפ ערך L קבוע לכל התמונה. המח שלנו מחפש תמונות שלמות ומשלים את הצורה, ומתעלם מהרקע.

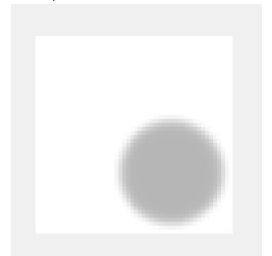
שאלה 8

(א)

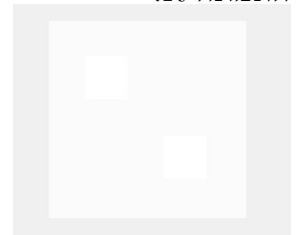
עבור $twoSquares(1)$ קיבלנו:



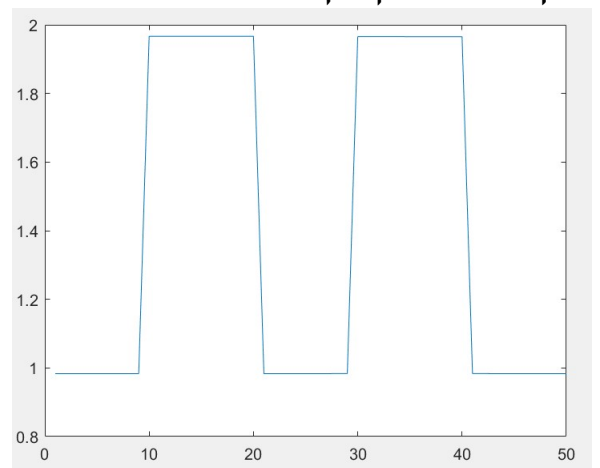
התמונה L שהתקבלה:



התמונה R :

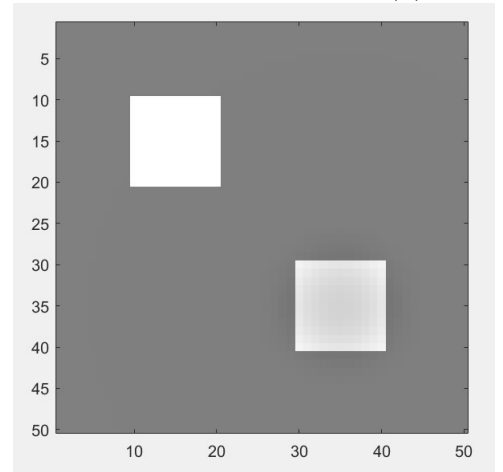


הגרף של האלכסון שקיבלנו:

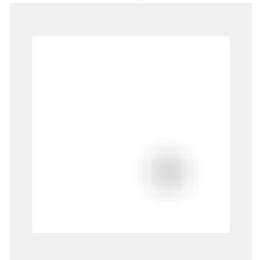


(ב)

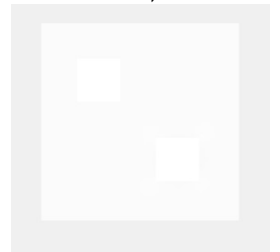
עבור $twoSquares(2)$ קיבלנו:



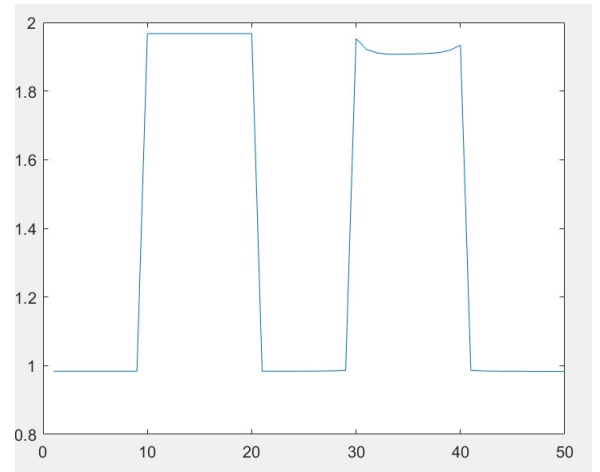
עבור L קיבלנו:



עבור R קיבלנו:



הגרף של האלכסון שקיבלנו:



הסבר:

כשאנו מסתכלים על האלכסון אנו רואים כי יש שינוי בין R של שתי התמונות. זה נובע מכך שההנחה הראשונה לא מתקיימת. כלומר יש כאן גרדיאנטים שנובעים גם מ $Reflectance$ וגם מ $Illumination$. לפי ההנחה הראשונה של אלגוריתם הרטינקס, כל שינוי הוא שינוי בגורם יחיד (תאורה או החזרה) אך לא בשניהם. בתמונה השנייה ההנחה הזו מופרת - השינוי בתאורה (ה"צל" האפור) והשינוי במקדם ההחזרה (הריבוע הלבן) מופיעים ביחד, ולא ניתן להפריד בצורה ברורה בין הגבולות שלהם. הדבר גורם להפרה של ההנחה ולביצועים פחות טובים של האלגוריתם.

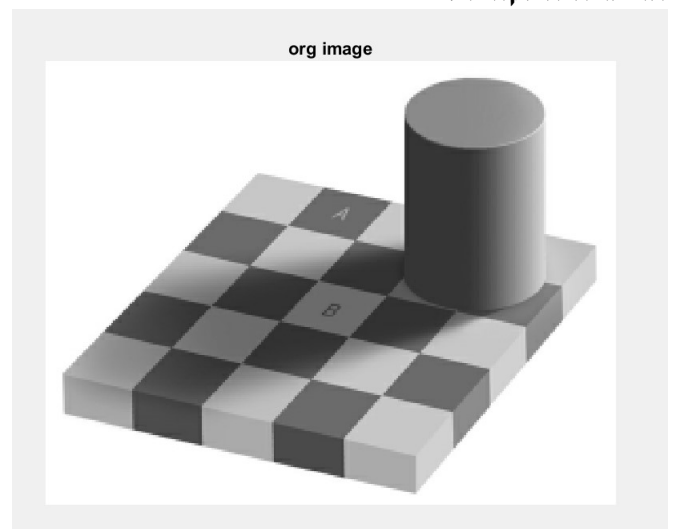
שינוי ה $Threshold$ לא ישנה את הגרף של 2, אלא רק הגרף של 1 השתנה. משום שהסף קובע את הנקודה בה מבחינים בין תאורה לבין החזרה - אך מכיוון שבמקרה זה שניהם משתנים יחד, בחירת סף שונה לא תשפיע.

שאלה 9

(א)

תחילה בדקנו וראינו כי הצבע של שתי המשבצות שווה ל 0.4196.

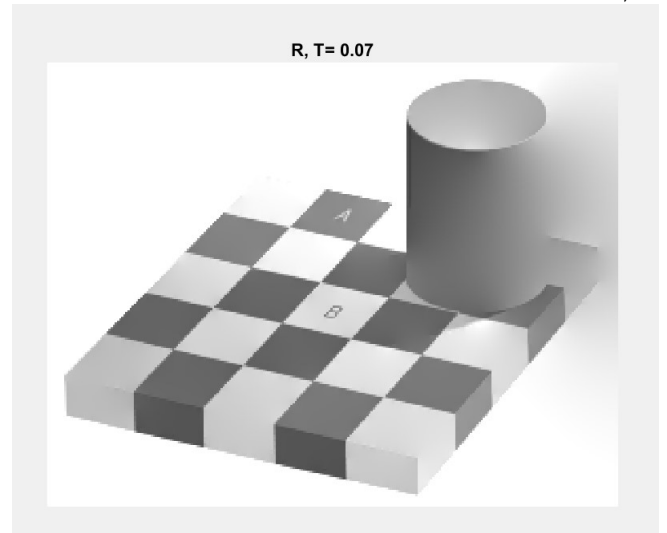
התמונה המקורית:



כאשר הדפסנו את R של שתי המשבצות קיבלנו כי:
התוצאה שקיבלנו היא :

calculated reflectance of the two checks: 0.5665 , 0.8768

כלומר, למרות ששתי הקוביות באותו הצבע, קיבלנו R שונה. זה מתכתב עם האשליה כי קוביה B נראית בהירה יותר מקוביה A .



(ב)

האלגוריתם נכשל כאשר:

המשבצות המוצללות הקרובות לגליל. במשבצות אלה, חל שינוי במקדם ההחזרה וברכיב התאור במקביל. בסתירה להנחת האלגוריתם כי רק אחת מהתופעות מתרחשת. האלגוריתם "מניח" שמדובר בשינוי במקדם ההחזרה בלבד, ולכן לא מחלץ אותו כראוי. אם נעלה את הסף, יותר שינויים חדים ייתפסו כשינויים ברכיב התאורה, ואז התופעה השלילית באזור זה תשתפר.

בצד הימני של הגליל - באזור זה מתרחשת "מריחה" אפורה על הרקע הלבן (ש"מאחורי" הגליל). מבחינת האלגוריתם, כנראה שהשינוי ההדרגתי בתאורה לרוחב הגליל ממשיך גם במעבר מהגליל אל הרקע הלבן. הדבר סותר את ההנחה של אלגוריתם לפיה שינויים הדרגתיים הם שינויי תאורה (במקרה הזה, השינוי ההדרגתי מהגליל אל הרקע הוא שינוי במקדם ההחזרה, שנתפס ע"י האלגוריתם כשינוי תאורה). אם נוריד את הסף, נעלה את הסיכוי ששינויים עדינים ייתפסו כשינויים במקדם ההחזרה, ובכך נשפר את ההפרדה של הגליל מהרקע.

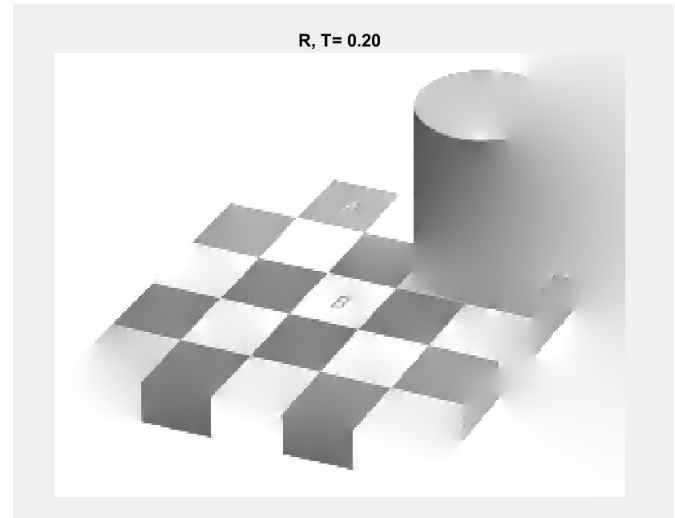
(ג)

שינוי ערך הסף - לא יעזור מכיוון ש:

1. אם נגדיל את ערך הסף ל 0.2:

נקבל את התוצאות הבאות:

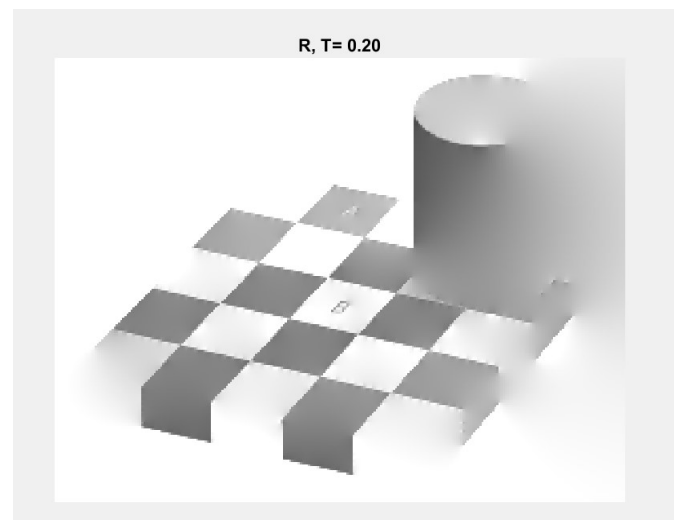
calculated reflectance of the two checks: 0.7365, 0.9695



2. אם נקטין את ערך הסף ל 0.02:

נקבל את התוצאות הבאות:

calculated reflectance of the two checks: 0.5067, 0.5819



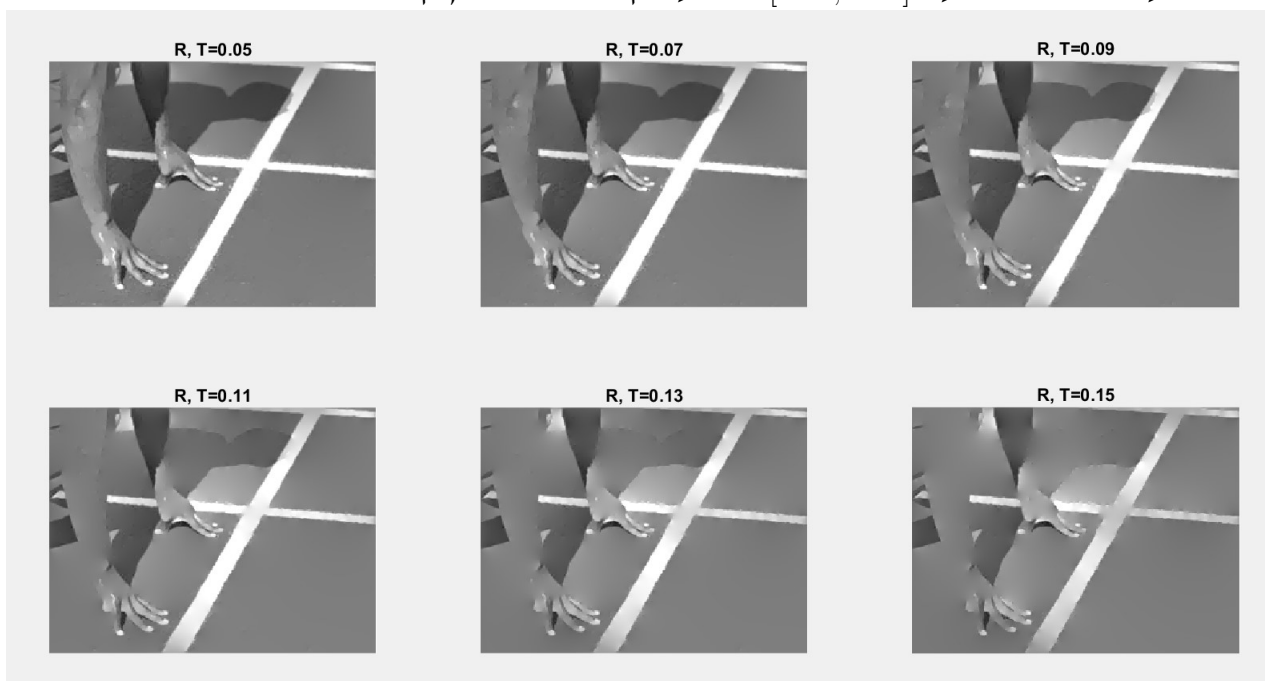
ראינו אם כן כי שינוי ערך הסף לא יפתור את הבעיה. משום שיש לנו טרייד אוף. העלאת הסף משפרת את הבעיה באזור הראשון אך מחמירה את הבעיה באזור השני, ולהיפך - מפני שהשיפורים שהיינו רוצים לבצע בכל אזור באים אחד על חשבון השני.

שאלה 10

התמונה המקורית היא התמונה הבאה:



לאחר הפעלת האלגוריתם עם $T = [0.05, 0.15]$ עם קפיצות בגודל 2, קיבלנו את התוצאות הבאות:



הסבר:

אנו רואים כי האלגוריתם לא מצליח להסיר את הצל בצורה הנכונה. וככל שהסף גדל - כך הצל נעלם מהרצפה, אך במקביל הידיים מתמזגות עם הרצפה.

בנוסף, הצל שמושפה מהאלגוריתם הינו הצל הרחוק יותר מהגוף. בעוד הצל הקרוב לגוף (הצל שנוצר מהיד) לא מושפע כלל. הסיבה לכשלון היא: הצל יוצר שינויים חדים בגבול הצל והרצפה, בשונה מהנחת האלגוריתם כי שינויים חדים לא נוצרים

מתאורה, אלא ממקדם ההחזרה.

לכן, האלגוריתם מפרש את השינויים הללו כשינויים במקדם ההחזרה, ולא כשינויי תאורה. בנוסף, גוון הצל והרצפה דומים ולכן אם אחד מהם משתנה, השני משתנה יחד איתם.

שינויים האלגוריתם לא ישפרו את המצב משום שהשינויים במקדם ההחזרה והתאורה מתנהגים באופן דומה. שינוי הסף משפיע רק על הנקודה הצל והיד מתמזגים עם הקרקע, אך הוא לא תורם להפרדה בניהם.

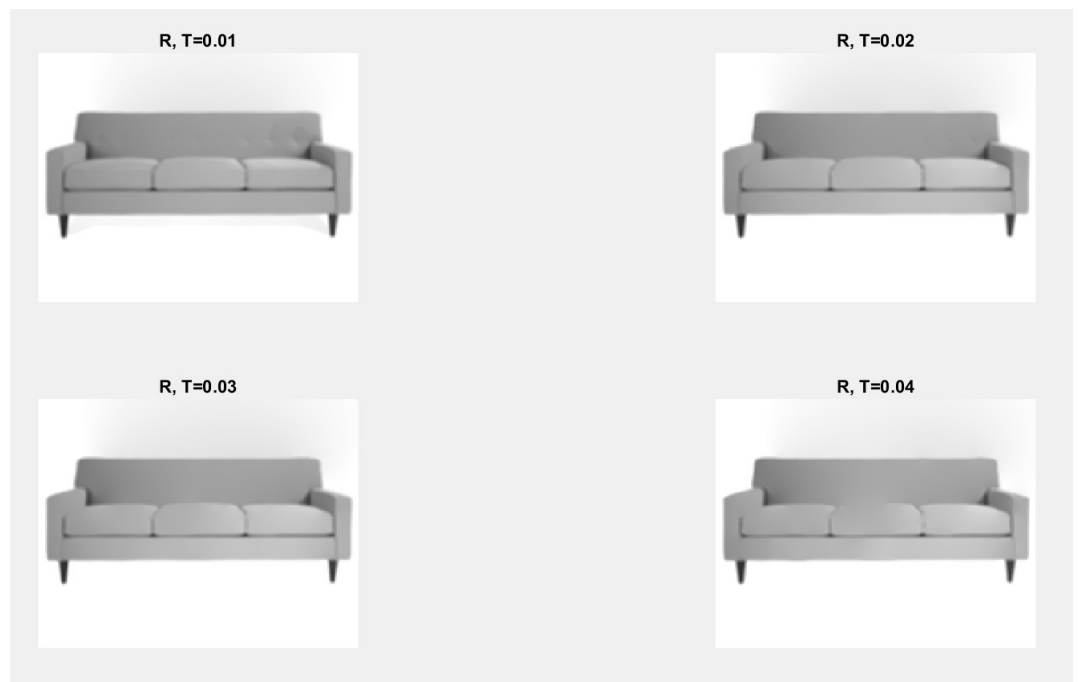
כדי לעזור לאלגוריתם, ניתן לסמן את מיקומו של האצן ומיקומו של הצל, באופן כזה כשיש הבדל ברור יש יותר סיכויי שהאלגוריתם יצליח לבצע את ההפרדה כראוי.

שאלה 11

התמונה המקורית:



התמונות לאחר הפעלת האלגוריתם:



הסבר:

ניתן לראות כי האלגוריתם אכן עובד טוב על התמונה הנוכחית. הצל שנמצא מתחת לספה נעלם אפילו כאשר T נמוך. אך נוצרת הילה מאחורי הספה, בנוסף הצל שנוצר ממשענות היד ומשענת הספה לא נעלם.

האלגוריתם עובד על התמונה הזאת טוב יותר, משום שהתמונה יותר מתאימה להנחות האלגוריתם. באמת השינויים החדים הם בגבולות הספה, ומתרחשים בגלל החזרי האור. בנוסף, ההנחה השניה מתקיימת ואין שינויי החזרה ותאורה שמתרחשים במקביל.

לגבי הצל שנוצר מידידת הספה, הוא לא נעלם משום שהתאורה יוצרת שינוי חד בניגוד להנחת האלגוריתם.

האלגוריתם מצליח על תמונה זו משום שהיא מתכתבת עם הנחות האלגוריתם, והיא טבעית יותר. בניגוד לתמונה הקודמת.