תרגיל Matlab מספר 3

הוראות כלליות

1. יש לשמור ולטעון תמונות איתם תעבדו **בספרייה אחת מעל** הספרייה בה נמצאים קבצי הקוד. לדוגמה:
   `>> I = imread('..\Lena.jpg');`
2. יש להימנע ככל האפשר משימוש בלולאות בקוד, ולעשות שימוש בכתיב מטריצי.
3. יש להגיש את כל התרגיל כסקריפט יחיד. ניתן ורצוי לממש שאלות או חלקים מהשאלות כפונקציות (לא לשכוח לצרף אותם כקבצים נפרדים!!).
4. יש להקפיד לתת כותרת מתאימה לכל תמונה או גרף, וגם לצירים כאשר זה רלוונטי. יש לדאוג שגרפים ותמונות יוצגו בגודל מספק להבנת תוכנם.
5. יש לכתוב כל סעיף ב-cell שונה. יוצרים cell חדש ע"י כתיבת התווים '%%' בתחילת שורה ריקה. (תווים אלו מבדילים בין cells). התווים הופכים להיות מודגשים ואז אפשר להפעיל בנפרד את ה-cell הנוכחי, הצבוע בצהוב, בעזרת Ctrl+Enter. לכל cell יש לרשום את מספר הסעיף ככותרת.
6. יש לתעד את הקוד באופן סביר ובמיוחד במקומות בהן מבוצעות פעולות לא טריוויאליות.
7. את התרגיל יש להגיש אלקטרונית **דרך Moodle**, בחלק של Matlab Assignments בצורה הבאה:
   יש לשמור את קבצי התרגיל שלכם ללא תמונות קלט/פלט כלשהן, בקובץ ZIP כך:
   ex3__<ID1>_<ID2>_**.zip**, כאשר <ID1> ו-<ID2> הינם מספרי הסטודנט של שני המגישים.
   לדוגמא: **ex3_012345678_987654321.ZIP** עבור תרגיל של מגישים בעלי מספרי הסטודנט 012345678, 987654321.
   ההגשה ב-moodle תעשה רק ע"י אחד מבני הזוג. יש לוודא כי גם שם קובץ ה-zip וכן גם העמוד הראשון של **קובץ ה-pdf** יכילו את **שני מספרי ת.ז.** של שני המגישים.

   קובץ ה-ZIP הנ"ל יכלול את הקבצים הבאים:

   א. סקריפט הרצת התרגיל וכל יתר הקוד הרלוונטי הנדרש להשגת כל התוצאות.
   ב. קובץ PDF יחיד עם תשובות לשאלות התיאורטיות יחד עם כל תמונה וגרף שהתקבלו כפלט מהקוד. יש לרשום את פרטי המגישים במסמך.
   ג. אין צורך להגיש את התמונות או את קבצי העזר שניתנו לכם.

8. **את התרגיל יש להגיש (מומלץ בזוגות) עד ליום ה' בתאריך 27.6.2019 בשעה 23:55.**
   על כל יום איחור בהגשה (שלא אושר מראש ע"י סגל הקורס) יורדו 4 נקודות אוטומטית. הגשה מאוחרת תעשה ישירות לאחראי התרגילים ולא דרך Moodle.
9. התייעצות עם חברים מותרת ואף מומלצת, אולם את הקוד עליכם לכתוב בצורה עצמאית. הסגל יתייחס בחומרה המקסימלית להעתקות.
10. מספר קבצי עזר ב-Matlab מצורפים באתר. כמו כן ניתן להיעזר כמובן ב-help של Matlab.
11. שאלות לגבי התרגיל ניתן להפנות לאחראי תרגילי ה-Matlab דרך הפורום המתאים ב-Moodle או במייל.

בהצלחה!

1. דגימה ושחזור

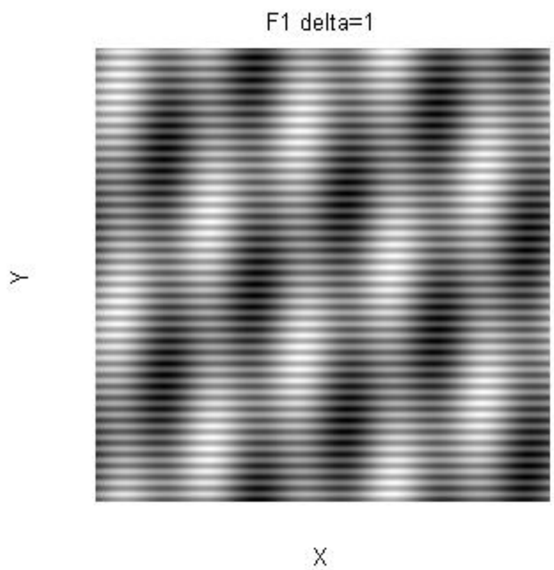
<u>הערה:</u> לאורך כל השאלה, הדגימה היא דגימה בשיטת השכן הקרוב.

הציבו את ערכי פרמטרים R ו-S בשאלה זו המתקבלים מהתשובות לשאלות הטריוויה הבאות:

R = הפרס על ראשו של Tuco (הידוע כ"מכוער"), בדולרים, לפני שהוסגר לראשונה (בעלילת הסרט "הטוב הרע והמכוער").

S = מספר הסוסים שמושכים את העגלה בה נמצא החייל Bill Carson (או מי שמתחזה לחייל בשם זה) בסרט "הטוב הרע והמכוער".

באיור למטה מוצגת דגימה במרווחי דגימה $\Delta x=\Delta y=1$ של הפונקציה F_1 (נסמן את הפונקציה באיור כ- $F_1^{\Delta=1}$), כאשר הדגימה הראשונה בכל ציר היא ב-0 (הפינה השמאלית העליונה בתמונה), ושאר הדגימות בערכים גבוהים יותר. פונקציה זו מהווה קומבינציה של חיבור ו/או כפל של שלושה סינוסים מהצורה $\sin\left(2\cdot\pi\cdot f_i\cdot i\right)$ בעלי התדרים הבאים $f_i\in\left\{\frac{20}{R},\frac{50}{R},\frac{400}{R}\right\}$, כאשר $i\in\left\{x,y,x+y\right\}$ (לאו דווקא בהתאמה), ואמפליטודה השווה ל-1. התמונה הינה בגודל $(R/10)\times(R/10)$.



א. מהי F_1?

ב. שחזרו את $F_1^{\Delta=1}$. הציגו את התמונה המתקבלת כתמונת גווני אפור מנורמלת (לא לשכוח לאפשר לפונקציה imshow לבחור את התחום הדינמי ע"י הוספת סוגריים ריקים []).

ג. הציגו את התמונה **במישור התדר** כתמונת גווני אפור (אמפליטודה בלבד). הסבירו את התמונה המתקבלת, <u>כללו בהסבריכם התייחסות למיקום האנרגיה בפיקסלים בתמונה **(מיקום מדויק)**, וכיצד הדבר מתיישב עם התיאוריה.</u>

תזכורת: כדי ליצור מטריצה של אמפליטודת התדר יש להפעיל טרנספורמציית פורייה דו-ממדית, לבצע fftshift ולקחת את הערך המוחלט של התוצאה.

ד. דגמו את הפונקציה F_1 במרווחי דגימה $\Delta x = \Delta y = S+1$ (לקבלת $F_1^{\Delta=(S+1)}$), כאשר הדגימה הראשונה בכל ציר היא ב-0, ושאר הדגימות בערכים גבוהים יותר, כך שתתקבל תמונה בגודל $\lceil R/[10\cdot(S+1)]\rceil \times \lceil R/[10\cdot(S+1)]\rceil$. הציגו את התמונה המתקבלת, וחיזרו על סעיף ג' עבור מקרה זה.

כעת נבחן כיצד תופעות אלו נראות בתמונה אמיתית:

ה. צלמו תמונה כלשהי ברחבי הטכניון באמצעות הסמארטפון. על התמונה להכיל פרטים רבים, כגון עצים, אנשים וכדומה. טענו את התמונה, הפכו אותה לתמונת רמות אפור (באמצעות rgb2gray), חתכו אותה לממדים של $(R/5)\times(R/5)$ והציגו אותה במישור התדר כתמונת גווני אפור (אמפליטודה בלבד). כדי לבחון גם את התדרים שהינם פחות דומיננטיים בתמונה, היעזרו באופרטור $\log_{10}(|A|+1)$ לצורכי תצוגה בלבד, כאשר A הינה מטריצת האמפליטודה.

ו. בצעו תת-דגימה לתמונה כך שתתקבל תמונה בגודל של $(R/20)\times(R/20)$ (יש לשמור כל פיקסל רביעי בכל אחד מכיווני הצירים). הציגו את התמונה החדשה במישור המקום ובמישור התדר (אמפליטודה בלבד– באמצעות האופרטור מסעיף ה'). השוו לתמונה המקורית לפי ההנחיות הבאות:

1. במישור המקום: מהם ההבדלים שניתן לראות בין שתי התמונות? מה מאפיין את האזורים בהם רואים הבדלים בין התמונות?
2. במישור התדר: מהו ההבדל המרכזי בין שתי תמונות התדר?

ז. העבירו את התמונה המקורית (לפני תת-הדגימה) במסנן גאוסי בגודל 5x5, עם סטיית תקן של 3, ואז בצעו תת-דגימה לגודל של $(R/20)\times(R/20)$. הציגו את התמונה אחרי תת-הדגימה במישור המקום ובמישור התדר (אמפליטודה בלבד– לפי סעיף ה'). השוו לתוצאות סעיף ו'. הסבירו את הסיבה להבדלים בין התמונות ואת העיקרון של התהליך אשר בצעתם.

2. דגימת וידאו

א. פתחו את הקובץ SpongeBob.mat, הקובץ מכיל מטריצה תלת ממדית בה שני הממדים הראשונים הינם השורות והעמודות בתמונה והממד השלישי הינו הזמן. צפו בסרטון באמצעות implay.

ב. מצאו מהו זמן מחזור הריצה של בוב-ספוג (ביחידות של מסגרות (frames) לסיבוב) ומכך חשבו מהו תדר הסיבוב. הציגו מחזור ריצה שלם.

ג. מהו תדר Nyquist לדגימה בזמן של סרטון זה? התייחסו לתדר הסיבוב כתדר המקסימלי.
<u>הערה:</u> על מנת לבאר את התהליך, ניתן להכפיל את התדרים שמצאתם ב- 20 מסגרות לשנייה (תדר ברירת המחדל להצגת הווידאו ב-implay), כך שיתקבל תדר ביחידות של סיבוב לשנייה.

ד. דגמו את הסרטון כך שבוב-ספוג ירוץ אחורה, מהו תדר הדגימה המתאים לכך? על מנת לראות את הסרטון טוב יותר, ניתן לשנות את מספר המסגרות לשנייה ב-implay.
הסבירו מה מתרחש בדגימה אשר גורם לבוב-ספוג לרוץ אחורה.

3. דחיסה משמרת של תמונות

בשאלה זו אין להשתמש בפונקציה entropy

א. כתבו פונקציה אשר מקבלת תמונת רמות אפור (מסוג uint8) ומחזירה את האנטרופיה שלה. חישוב האנטרופיה ייעשה ע"י הנוסחה:

$$H(im) = -\sum_j P_j \log_2 P_j$$

כאשר P_j זהו ההסתברות להופעת הערך j, כלומר מספר ההופעות של הערך j בתמונה חלקי מספר הפיקסלים הכולל בתמונה. רמז: בנו היסטוגרמה על-מנת לחשב את ההסתברויות.

ב. טענו את התמונה heisenberg.jpg. חשבו את האנטרופיה של התמונה בעזרת הפונקציה שכתבתם בסעיף א'.

ג. **כתבו פונקציה** `[code,dict,avglen] = huffcoding(im)` שמבצעת קידוד Huffman על התמונה `im` ומחזירה את הווקטור הבינארי `code`, את המילון המתאים לו `dict` ואת קצב המידע הממוצע במילון `avglen`. לצורך כך, הפכו את התמונה לווקטור עמודה, בנו עבורו היסטוגרמה והשיגו וקטור של ערכים ייחודיים בו באמצעות הפונקציה `unique`. אז, השיגו מילון Huffman באמצעות הפונקציה `huffmandict`, וקודדו את וקטור העמודה באמצעות הפונקציה `Huffmanenco`.

ד. בנו קוד Huffman לתמונה שטענתם, וחשבו את אורך הקוד שהתקבל עבור התמונה. מהו יחס הדחיסה המתקבל?

ה. פענחו את הקוד באמצעות הפונקציה `huffmandeco`. חשבו את ה-MSE (שגיאה ריבועית ממוצעת) ביחס לתמונה המקורית. הסבירו את התוצאה.

ו. כעת נתנסה בקידוד הפרשים:

1. טענו את התמונה sunset.jpg והעבירו אותה לווקטור עמודה באמצעות שיטות הסידור הבאות:
   - סידור עמודה רגיל (Column stack).
   - סידור Hilbert – לשם כך היעזרו בפונקציה hilborder.m המצורפת. לקריאה נוספת: https://en.wikipedia.org/wiki/Hilbert_curve
2. עבור כל סידור, חשבו וקטור הפרשים (עם הפרש של 1) באמצעות הפרש אחורי (האיבר הראשון בווקטור יישאר ללא שינוי) והפכו אותו למטריצה בעלת הממדים המקוריים של התמונה. הציגו את ההיסטוגרמה המתקבלת בכל שיטת סידור. הסבירו את התוצאות.
   הערה: שימו לב לטיפוסי משתני הווקטורים בטרם חישוב ההפרשים.
3. בנו קוד Huffman עבור כל תמונת הפרשים באמצעות הפונקציה מסעיף ג'. בחנו את קצב המידע הממוצע שהתקבל בכל שיטה. איזו מבין שיטות הסידור עדיפה ולמה? הסבירו.

4. קוונטיזציה וקטורית אופטימלית

הצעות לפונקציות שימושיות בשאלה זו: reshape, unique, randperm, repmat, permute, kmeans.

א. ממשו קוונטייזר Max-Lloyd לווקטורים עבור תמונות צבע. המימוש יעשה ע"י פונקציה ובה המשתנים:

data – מטריצה תלת-ממדית אשר מייצגת תמונת צבע (RGB).

levels – מספר וקטורים מייצגים רצוי (גודל המילון).

meps – השיפור היחסי המינימלי.

dataout – מטריצה תלת-ממדית המכילה את התמונה לאחר קוונטיזציה.

distortion – וקטור באורך של מספר האיטרציות, המכיל את העיוות הממוצע שהתקבל בכל איטרציה (עפ"י שגיאה ריבועית).

QL – מטריצה בגודל (levelsX3) המכילה את וקטורי המילון (codebook).

חשבו את העיוות הממוצע D_m בכל איטרציה ואת ערך השיפור היחסי $\varepsilon = \left|D_m - D_{m+1}\right| / D_m$. קבעו את תנאי העצירה להיות $\varepsilon \le meps$.

<u>הדרכה:</u> לשם המימוש ניתן להיעזר בחוברת Image Processing Booklet המופיעה ב- Moodle. קוונטיזציית צבע ווקטורית מוסברת בפרק 4 החל מעמוד 71, ובעמודים 72-73 מוסבר האלגוריתם. שימו לב שאין לבצע קוונטיזציה לכל ערוץ בנפרד, אלא להתייחס לפיקסלים כנקודות במרחב תלת ממדי (R,G,B) כמצוין לעיל. יש להשתמש במטריקה אוקלידית. שימו לב שבניגוד למימוש עבור תמונה דו-ממדית, כאן מקטעי ההחלטה הם למעשה אזורי החלטה, ומוצאים אותם באמצעות ערכי התמונה עצמם.

<u>הערות לכל סעיפי שאלה זו:</u>

1) על הערכים של dataout להיות שלמים.

2) כאשר אתם מאתחלים את הקוונטייזר, בחרו את רמות האתחול בצורה אקראית, אך ודאו שהן שייכות לצבעי התמונה המקורית (data) ואין רמות זהות.

3) על מנת לממש אלגוריתם איטרטיבי תצטרכו לממש לולאה. קבעו מראש מספר איטרציות מרבי הגיוני, כך שהקוד יסיים לרוץ מתישהו גם אם יש לכם טעות בתנאי העצירה.

<u>טיפים למימוש:</u>

- על מנת למצוא שיוך של פיקסלים לרמות ייצוג (QL), מומלץ להשתמש בפקודות repmat (להגדלת מטריצות ע"י שכפולן) ו-min (למציאת מרחק מינימלי). שימו לב שתוצאת פעולת ה-min יכולה לשמש אתכם גם לחישוב רמות הייצוג הבאות.
- יש לשמור בין איטרציות רק אובייקטים רלוונטיים (על מנת למנוע בעיה של עומס על הזיכרון וכמו כן תרגול של בניית קוד נכון יותר לאלגוריתמים "סקלביליים" - scalable).

ב. הפעילו את הקוונטייזר על התמונה colorful.jpg, ובצעו קוונטיזציה ל-6 צבעים ול-15 צבעים כאשר האתחול נעשה בצורה אקראית. חשבו את העיוות הממוצע D_m (ע"פ שגיאה ריבועית) בכל איטרציה ואת השיפור היחסי ε. קבעו את תנאי העצירה להיות meps = 0.02. הציגו גרף של העיוות הממוצע כתלות במספר האיטרציות ואת התוצאות.

ג. חזרו על סעיף ב' מספר פעמים עבור 6 צבעים (אין צורך להציג את התוצאות). האם בכל פעם מתקבלת אותה תוצאה? הסבירו.

ד. חזרו על סעיף ב' כאשר האתחול הוא בצבעי היסוד אדום, ירוק, כחול, סגול, צהוב, תכלת, שחור, לבן, אפור:

[0.9 0 0], [0 0.9 0], [0 0 0.9], [0.6 0 0.6], [0.7 0.7 0], [0 0.8 0.8], [0 0 0], [1 1 1], [0.5 0.5 0.5]

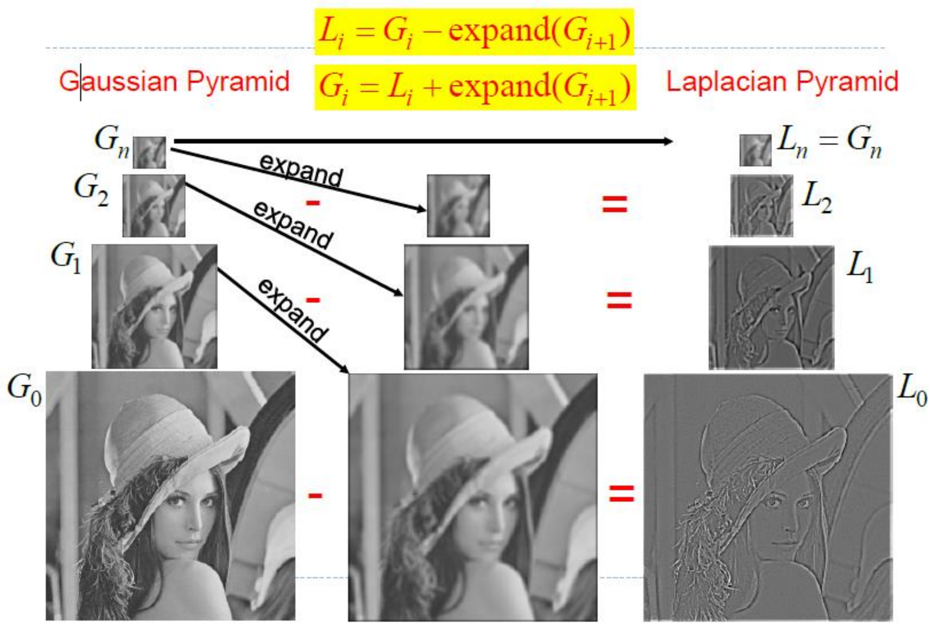
ה. השוו את התוצאות בשתי השיטות. היכן התקבלה תוצאה טובה יותר? נסו להסביר מדוע. השוו את כמות האיטרציות שנדרשו ע"מ להגיע לתוצאה בשתי השיטות. על מה זה מעיד?

5. ריבוי רזולוציות

הצעות לפונקציות שימושיות בשאלה זו: impyramid, imgaussfilt, rgb2gray, im2double, cell, immse.

בשאלה זו נתנסה בבניית פירמידה גאוסיאנית ולפלסיאנית, נבצע שחזור תמונה מתוך הפירמידה הלפלסיאנית, ונבצע היתוך תמונות (Image fusion) ע"י שימוש בריבוי רזולוציות.

בניית הפירמידות הגאוסיאנית והלפלסיאנית מתוארת באיור הבא:



- כל רמה i בפירמידה הגאוסיאנית (העמודה השמאלית באיור) נוצרת ע"י סינון גאוסי של הרמה ה-$(i-1)$ ודגימת פיקסל אחד מכל שני פיקסלים בכל ציר (reduce). הרמה ה-0 היא התמונה המקורית. בפקודת הסינון יש להוסיף את הארגומנט 'symmetric' כדי להמנע מאפקטי קצוות. יש להשתמש במסנן גאוסי בגודל 5X5 ובעל סטיית תקן של 1.
- כל רמה i בפירמידה הלפלסיאנית (העמודה הימנית) ע"י הרחבה של הרמה הגאוסיאנית ה-$(i+1)$ (expand) והחסרתה מהרמה הגאוסיאנית ה-i. הרמה הלפלסיאנית ה-n זהה לרמה הגאוסיאנית ה-n. הפירמידה הלפלסיאנית מחלקת למעשה את התמונה לתחומי תדרים שונים, כאשר התדרים הנמוכים ביותר מופיעים ברמה ה-0.

א. ממשו פונקציה בשם pyrGen אשר בונה פירמידה גאוסיאנית ולפלסיאנית. הפונקציה תקבל כקלט תמונת רמות אפור ומספר הרמות הנדרשות n (לא כולל הרמה ה-0) ותחזיר כפלט שני cells – אחד מכיל את $\mathrm{n}+1$ הרמות הגאוסיאניות של התמונה והשני את $\mathrm{n}+1$ הרמות הלפלסיאניות של התמונה.

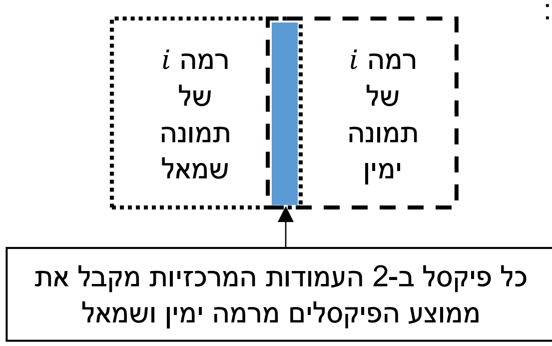
<u>הדרכה:</u> היעזרו בפונקציית impyramid המצורפת לשאלה, שבניגוד לפונקציה המובנית ב-MATLAB, ביצוע reduce ואח"כ expand לא משנה את גודל התמונה. רצוי לממש את הפונקציה בצורה רקורסיבית, כאשר לצורך כך ניתן להוסיף עוד משתנים לקלט.

ב. טענו את התמונות apple.jpg ו-orange.jpg, המירו אותם לרמות אפור ול-double באמצעות im2double. בנו לשתיהן פירמידה גאוסיאנית ולפלסיאנית עם $\mathrm{n}=4$ רמות באמצעות pyrGen. הציגו את התוצאה ונתחו את ההבדלים בין התמונות.

ג. ממשו פונקציה בשם imRecon אשר מבצעת שחזור לתמונה באמצעות הפירמידה הלפלסיאנית שלה. הפונקציה תקבל כקלט cell של הפירמידה הלפלסיאנית ותחזיר כפלט את התמונה המשוחזרת.
<u>הדרכה:</u> יש להתחיל מהרמה ה-n ולבצע לה expand. את התוצאה יש לסכום עם הרמה ה-$(\mathrm{n}-1)$, ולתמונה המתקבלת לבצע expand. את התוצאה יש לסכום עם הרמה ה-$(\mathrm{n}-2)$, וכן הלאה.

ד. שחזרו את apple.jpg ו-orange.jpg (לאחר ההמרה שביצעתם בסעיף ב') באמצעות imRecon והציגו את התוצאה ואת תמונת ההפרשים בין התמונה המקורית (ברמות אפור) למשוחזרת. נתחו את התוצאה וחשבו את שגיאת ה-MSE בין המקור לשחזור.

ה. כעת בצעו היתוך בין התמונה orange (להלן "תמונה ימין") לבין apple (להלן "תמונה שמאל") כפי שמתבצע במאמר של Burt & Adelson מ-1983:
http://www.cs.princeton.edu/courses/archive/fall05/cos429/papers/burt_adelson.pdf
<u>הדרכה:</u> תחילה בנו פירמידה לפלסיאנית לכל אחת מהתמונות, ואז בנו פירמידה לפלסיאנית לתמונה המשולבת באופן הבא: נסמן את ממדי הרמה ה-i של שתי התמונות ב-$2N$x2N. שרשרו את $N+1$ העמודות הימניות של הרמה ה-i בפירמידה של תמונה ימין לצד $N+1$ העמודות השמאליות של הרמה ה-i של תמונה שמאל. בצעו את השרשור כך שתהיה חפיפה של 2 העמודות השמאליות של הרמה הימנית עם 2 העמודות הימניות של הרמה השמאלית, ובאזור החפיפה כל פיקסל יקבל את הערך הממוצע בין הפיקסלים החופפים של שתי הרמות. חזרו על תהליך זה לכל $i=0\ldots n$.
להלן איור המדגים את בניית הפירמידה:



לבסוף, חברו את כל הרמות של הפירמידה החדשה באמצעות imRecon והציגו את התמונה שקיבלתם.

ו. בצעו שוב היתוך, הפעם ע"י חיתוך התמונות עצמן באופן פשוט ושרשור חצאי התמונות זו לצד זו. השוו את התוצאה שהתקבלה באזור המעבר שבין חצאי התמונות בסעיף זה לעומת הסעיף הקודם.