

# מבוא לעיבוד ספרתי של תמונות

## עבודה 2

מגישים:

איילה ראובן 314077033

ליאור עבדייב 206087611

### שאלה 1 – Edge detection

1.1. קראנו את התמונה "cameraman.tif", ונרמלנו אותה בין הערכים [0,1]:



### 1.2 Prewitt edge detector

1.2.1. בסעיף זה כתבנו פונקציית dip\_prewitt\_edge, שמוצאת את השפות באמצעות השיטה

prewitt. את מציאת השפות מימשנו על ידי קונבולוציה של התמונה עם המטריצות הבאות:

$$G_{Px} = \frac{1}{6} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, G_{Py} = \frac{1}{6} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

פעולה זו מממשת נגזרת אופקית ואנכית מה שמנחית אזורים בתמונה בהם יש שינויים קלים ולעומת זאת מחזק את האזורים בהם יש שינויים משמעותיים.

לאחר מכן נחשב את המוצא הסופי באמצעות החישוב:

$$\sqrt{G_x^2 + G_y^2}$$

כאשר  $G_x$  מוצא הקונבולוציה של התמונה עם  $G_{p_x}$  ו  $G_y$  מוצא הקונבולוציה עם  $G_{p_y}$ .

### 1.2.2.

נציג את פלט הפונקציה עבור שני ערכי "threshold" שונים:

Threshold = 0



Threshold = 0.2



עבור הערך 0 ניתן לראות יותר "מריחות" בתמונה מכיוון שערכי הנגזרת הם לא 0 (אמנם לא אזורים בתמונה עם שינויים מהותיים אבל כן שינויים בעלי שיפוע נמוך שהוא לא 0).

לעומת זאת עבור 0.2 אנו רואים שפות יותר חדות מכיוון שחתכנו את השיפועים הנמוכים שגורמים למריחה של התמונה.

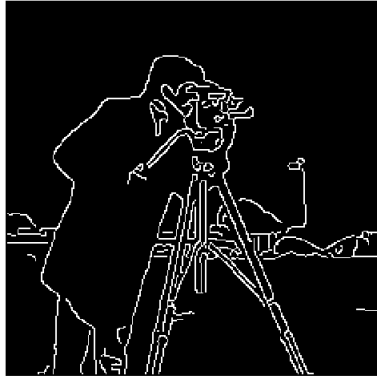
## 1.3 Canny edge detector

### 1.3.1 קראנו לגבי פונקציית edge('I','canny') של מטלב וענינו על השאלות:

על מנת למצוא את ערכי ברירת המחדל של הפונקציה הפעלנו את הפונקציה בצורה דיפולטיבית ועל ידי שינוי ערכי הthreshold והגענו לשתי תמונות זהות על ידי ערכי הthreshold: [0.0313,0.078].

**1.3.2.** נפעיל את הפונקציה עם שני ערכי threshold שונים ונקבל את התמונות הבאות:

canny with [0.1 0.2] threshold



canny with default parametres



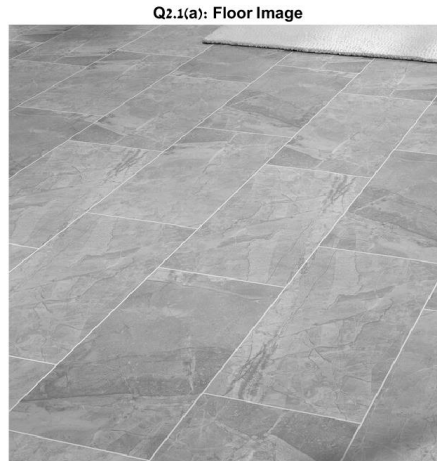
אם נתבונן בתמונות נבחין שעבור הערכים הדיפולטיים ניתן להבחין ביותר שפות אבל מצד שני לא הייתי מגדיר את הערכים הדיפולטיים כאופטימליים מכיוון שאנו רואים שפות שממקומות באדמה שלא היינו רוצים שיחשבו כשפות בתמונה. לעומת זאת עבור הערכים [0.1 0.2], המוצא פחות רגיש לשינויים אבל מצד שני קיבלנו שפות יותר "אמיתיות".

**1.3.3.** הגענו לשאלה אחרי שכבר שיחקנו עם פרמטרים שונים ולכן קיבלנו שהערכים הדיפולטיים דווקא פחות טובים.

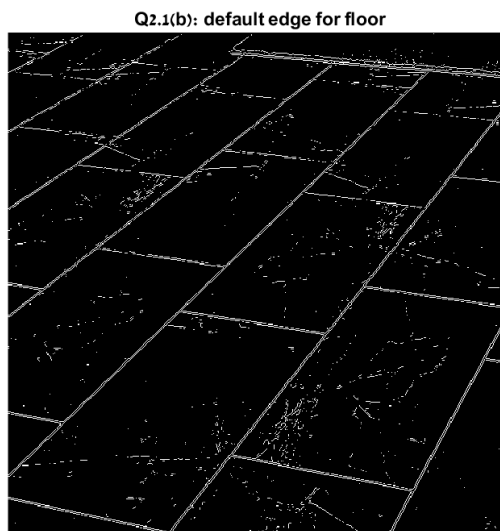
## שאלה 2 – Hough Transform:

### Hough line Transform 2.1

(a) התבקשנו לקרוא את התמונה הבאה, להפוך אותה לסקאלה של אפורים ולנרמל אותה :



(b) מצאנו את הקצוות של התמונה ע"י שימוש בפונקציה edge של matlab. התוצאה :



נשים לב שקיבלנו בלבן את המעברים בין המרצפות בצורה חדה, זאת מכיוון שהם בצבע לבן וערכם גבוה, כך שהם מהווים את הקצוות הבולטים בתמונה. בנוסף מכיוון שקיימים גם מעברים חדים בצבעים שבתוך הקרמיקה במרצפות נקבל פסים וקצוות גם בתוך המרצפות עצמן.

ה-edge detector לפי ברירת המחדל של מאטלב שהפונקציה edge הולכת לפיו הוא sobal edge detection model.

הפרמטרים ברירת המחדל הם :

I- התמונה שנרצה למצוא את הקצוות שלה.

Method- לפי איזה שיטה נבצע את גילוי הקצוות. ברירת המחדל היא sobal.

Threshold- ערך הסף שבו יעבוד האלגוריתמים לזיהוי השפות. ערך הסף קובע אם הפקסל הוא קצה או לא- כלומר אם הפיקסל יקבל את הערך 0 או 1. אם הוא לא מוגדר ע"י המשתמש הוא ישתנה באופן אוטומטי.

Direction- הכיוון בו נגלה את הקצוות, כלומר כיוון הגזירה. פרמטרי ברירת המחדל הם בכיוון האופקי והאנכי.

(c) התבקשנו לכתוב את הפונקציה `dip_hough_lines`, שמקבלת תמונה של קצוות וערכי קוונטיזציה  $R_0$  ו-  $\theta_0$  את מטריצת הough של התמונה. חישבנו את המטריצה עבור הערכים  $(R_0, \theta_0) = (1, 1)$  והערכים  $(R_0, \theta_0) = (5, 4)$  לפי האלגוריתם הנתון. הסבר על האלגוריתם:

א. כפי שלמדנו בהרצאה ניתן לבטא ישר במישור באמצעות הנוסחה הבאה:

$$\rho = x \cos(\theta) + y \sin(\theta)$$

$$\rho \in [-\sqrt{\text{rows}^2 + \text{col}^2}, \sqrt{\text{rows}^2 + \text{col}^2}] \quad \text{ו-} \quad \theta \in [-90, 90]$$

בקפיצות של  $\rho_0, \theta_0$ .

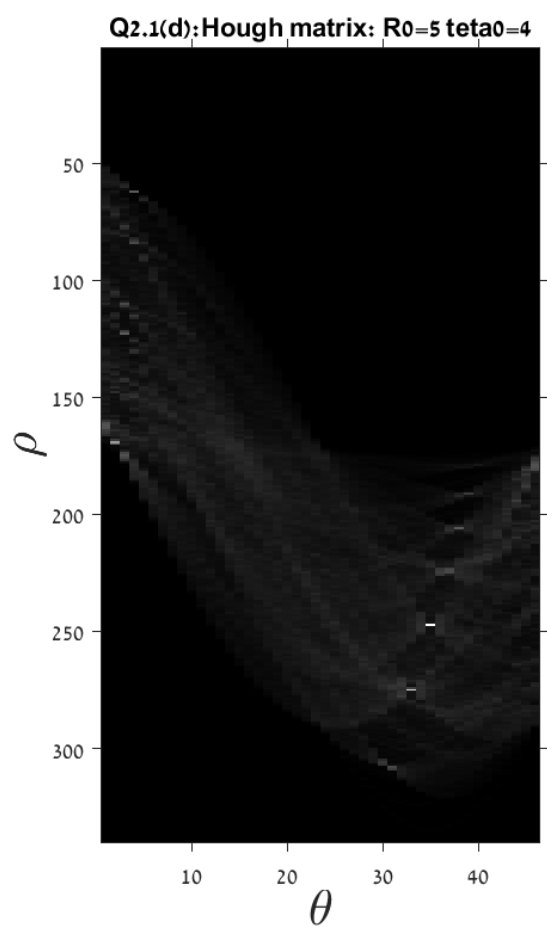
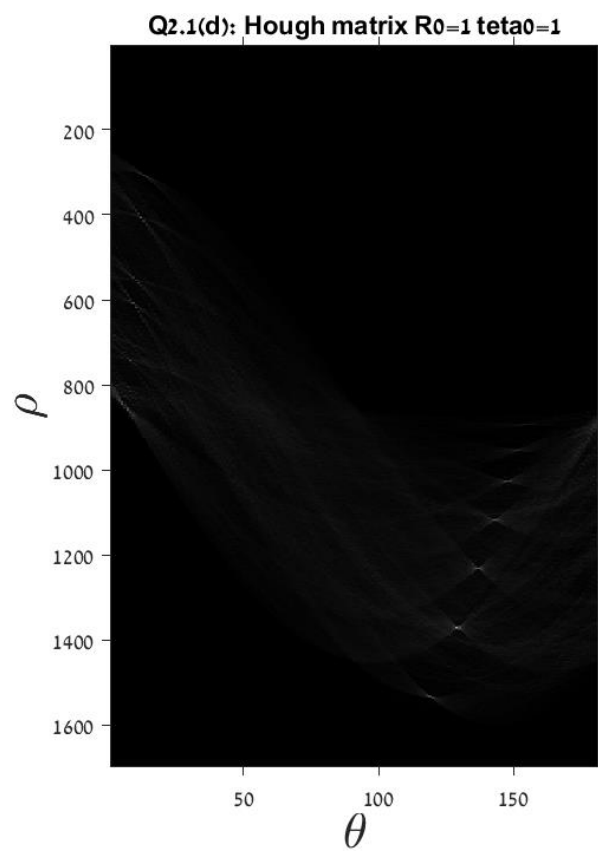
ב. באמצעות הפונקציה `find` נמצא את כל הנקודות  $(x, y)$  במוצא פונקציית `edge` בהן הערכים לא התאפסו.

ג. נעבור כל נקודה ונבנה עבורה גרף פונקציה  $\rho(\theta)$  עבור ערכי  $\rho, \theta$  בתחום שצוין בסעיף א.

ד. במטריצת `ough`, נוסיף 1 לפיקסלים המתאימים  $(\rho(\theta), \theta)$ .

ה. לאחר שנבצע את אותה פעולה עבור כל הנקודות  $(x, y)$ , נקבל מטריצה שמכילה את הסכימה של כל הנקודות, נתבונן במטריצה ונראה את הנקודות בהן עוברים הכי הרבה גרפים של סינוס, אלה נקודות הצומת.

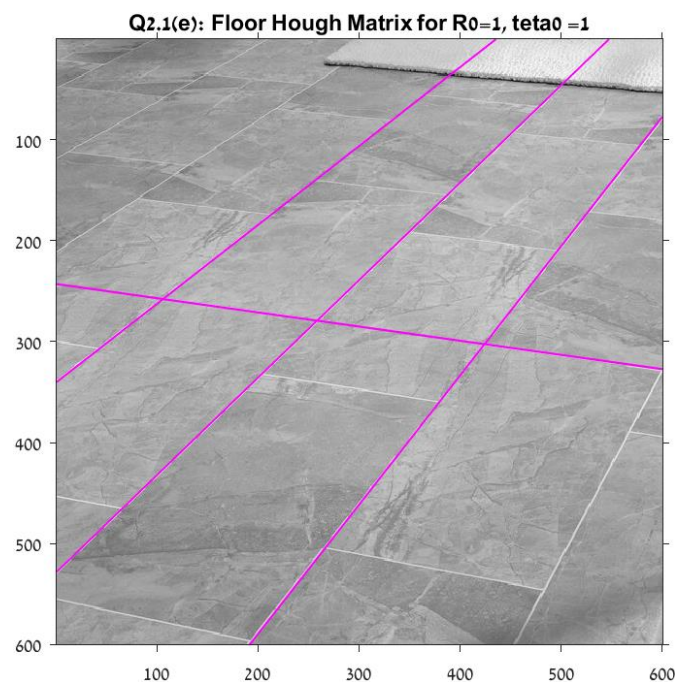
(d) התבקשנו להציג את המטריצות שקיבלנו בסעיף הקודם. התוצאות בעמוד הבא:

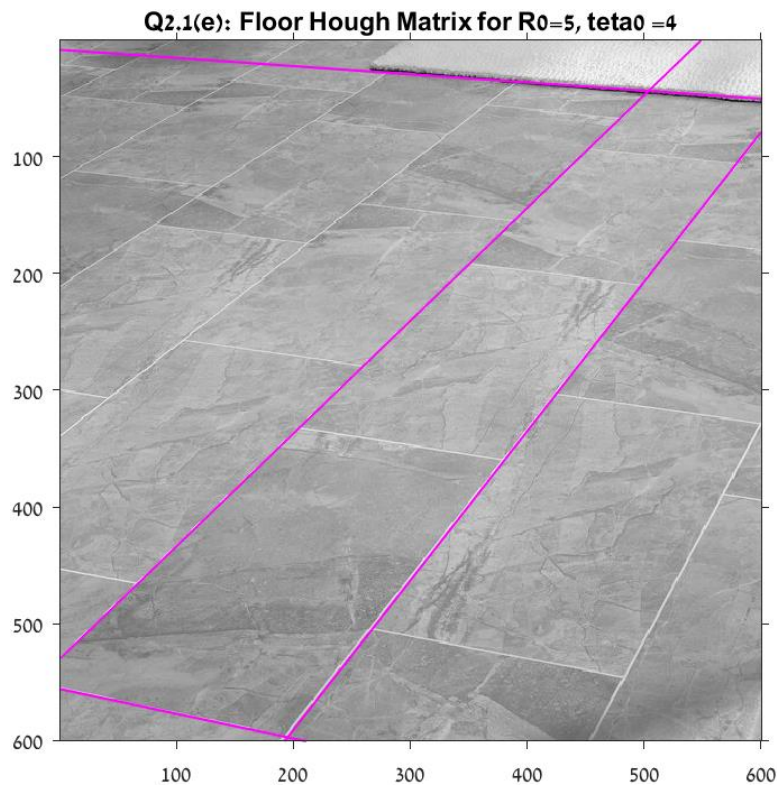


ניתן לראות שהתקבלו סינוסים. בנקודות החיתוך שלהם קיבלנו נקודות לבנות חזקות, כלומר בנקודות האלה קיבלנו ערכים גבוהים. בנקודות הלבנות ביותר עוברים הכי הרבה סינוסים, לכן אלו יהיו הנקודות המשמעותיות ביותר שמייצגות את הישרים המשמעותיים בתמונה המקורית. את הנקודות האלה נרצה למצוא כשנחפש את הקווים המשמעותיים ביותר בתמונה.

עבור ערכי  $\theta_0, R_0$  קטנים יותר נקבל סינוס רציף יותר וברור יותר ועבור ערכים גדולים יותר נקבל סינוס יותר "מטושטש", אך התחומים בהם נמצאות הנקודות הלבנות יותר נשארו אותו הדבר פחות או יותר. אפשר לראות גם שעבור ערכי הקוונטיזציה הגבוהים יותר הנקודות הלבנות חזקות ובוולטות יותר.

(e) התבקשנו למצוא את ארבעת הקווים המשמעותיים ביותר בתמונה ולהציג אותם על התמונה. ביצענו זאת עבור שתי הקוונטיזציות מהסעיפים הקודמים. על מנת למצוא את נקודות הפיק השתמשנו בפונקציה של מאטלב *houghpeaks* (בחרנו threshold של 1 כדי לא לפספס קווים). התוצאות שקיבלנו:





(f) בשני המקרים זיהינו את הקווים הארוכים ביותר כקווים דומיננטיים וההבדלים התקבלו בשני הקווים האחרים.

עבור ערכי קוונטיזציה נמוכים יותר הפונקציה שלנו רגישה יותר. המטריצה שלנו תהיה בעלת ערכים שונים רבים יותר ולכן נקבל פחות תאים במטריצה עם אותו הערך ולכן יהיו פחות נקודות מרכזיות משותפות ומכאן נקבל פחות פיקים משמעותיים. כתוצאה מכך קו אחד עבה בתמונה עלול להיות מזוהה כשני קווים נפרדים ולכן אנחנו עלולים לפספס אותו כקו משמעותי. מצד שני נקבל עבור ערכים אלו קווים מדויקים מאוד ונוכל לזהות קווים דקים יותר.

בתמונה אפשר לראות שפספסנו קו משמעותי כמו הקו של השטיח בתמונה, לעומת זאת אפשר לראות ששני קווים עם הפרדה זוהו כקו אחד- כלומר למרות ההפרדה זיהינו את הקו כקו משמעותי.

עבור ערכי קוונטיזציה גבוהים קיבלנו במטריצה נקודות לבנות חזקות יותר, לכן נקבל פיקים ברורים יותר בנקודות החיתוך המשמעותיות. במילים אחרות נקבל פונקציה פחות רגישה ולכן ריכוז גבוה של אותם הערכים בנקודות המשמעותיות ולכן נוכל לזהות קו עבה כקו אחד ולא כשני קווים. אבל גם מכיוון שהקפיצות גדולות יותר נפספס יותר ערכים ולכן פחות נצליח לזהות קו דומיננטי שמתפצל לקווים קטנים יחסית כמו שקיבלנו עבור הקוונטיזציה הגבוהה או קו שהוא דק יחסית ורחוק אך דומיננטי. בתמונה הצלחנו לזהות את הקו של השטיח שלא הצלחנו לזהות קודם, אך לא הצלחנו למצוא את הקו שמופיע בשתי מרצפות עם הפרדה כקו אחד דומיננטי. בנוסף מכיוון שהקוונטיזציה גסה פספסנו קו ארוך נוסף שמקביל לשניים שמופיעים בשתי התמונות.



## Hough Circle Transform 2.2

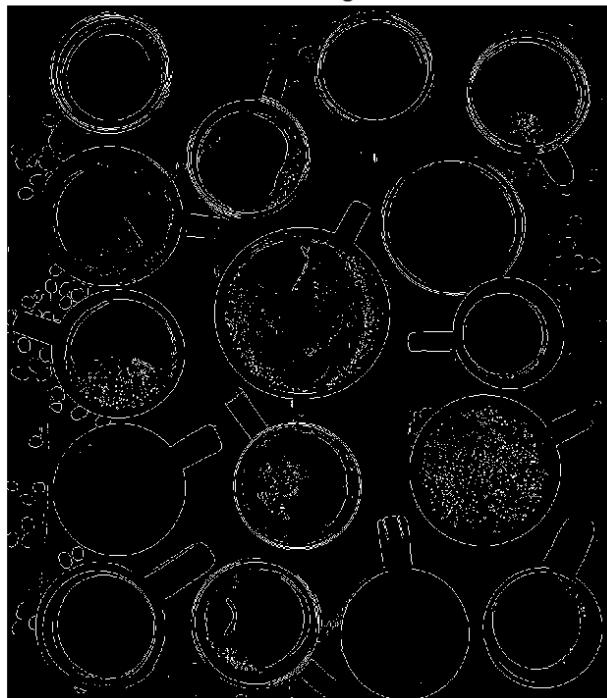
(a) התבקשנו לקרוא את התמונה של כוסות הקפה, להעביר אותה לסקאלה של אפורים ולנרמל אותה. התוצאה:

Q2.2(a): Coffee Image



(b) כמו בשאלה הקודמת מצאנו את הקצוות בתמונה ע"י הפונקציה *edge* של מאטלב. התוצאה:

Q2.2(b): default edge for coffee



אפשר לראות שהקצוות העיקריות שקיבלנו בתמונה אלה השפות של ספלי הקפה. בנוסף אפשר להבחין בקצוות של הקצף שבתוך הקפה וגם של הפולים.

(c) התבקשנו לכתוב את הפונקציה *dip\_hough\_circle* שמחשבת את מטריצת Hough למציאת מעגלים בתמונה. הפונקציה מקבלת תמונה של קצוות וערכי קוונטיזציה  $R_0$  ו- $\theta_0$  ומחשבת את מטריצת hough

של התמונה. חישבנו את המטריצה עבור הערכים  $(R_0, \theta_0) = (1,1)$  והערכים  $(R_0, \theta_0) = (4,10)$  לפי האלגוריתם הנתון בשאלה. הסבר:

נרצה לייצג מעגלים, ומעגלים נייצג ע"י קואורדינטות של מרכז המעגל  $x$  ו- $y$  ורדיוס המעגל  $R$ . לכן נבנה מטריצה תלת ממדית.

תחילה יצרנו מטריצת אפסים ממימד  $M \times N \times |R|$ , יצרנו וקטור של רדיוסים  $R_0: 100$ : 80 ווקטור של זווית  $\theta_0: 360$ : 0. ואז לכל פיקסל לבן בתמונה של הקצוות חישבנו את המעגל אותו הוא מגדיר, והוספנו 1 במקום המתאים במטריצת ה-*hough* שלנו. ערך של כל תא במטריצה מתאר את מספר הנקודות הלבנות בתמונת השפות שבמרחק  $R$  מהתא. כלומר עבור כל נקודה לבנה בתמונה של הקצוות נעבור על כל המעגלים ברדיוסים הנתונים שעוברים דרכה ונוסיף 1 במטריצה במקום המתאים.

(d) התבקשנו לחשב את הזמן שלוקח לנו לחשב את המטריצה עבור הקונטיזציה  $(R_0, \theta_0) = (1,1)$  ולמצוא ערכי קונטיזציה אחרים שעבורם נקבל את אותו הדיוק אך בפחות זמן. עבור הקונטיזציה  $(R_0, \theta_0) = (1,1)$  קיבלנו שלוקח לנו 29.12 שניות לחשב את מטריצת ה-*hough*. אחרי כמה ניסיונות מצאנו שעבור ערכי הקונטיזציה  $(R_0, \theta_0) = (1,5)$  נחשב את המטריצה תוך 6.84 שניות כלומר הרבה יותר מהר (נראה בהמשך שגם קיבלנו את אותו הדיוק- כלומר את אותם המעגלים הדומיננטיים בתמונה).

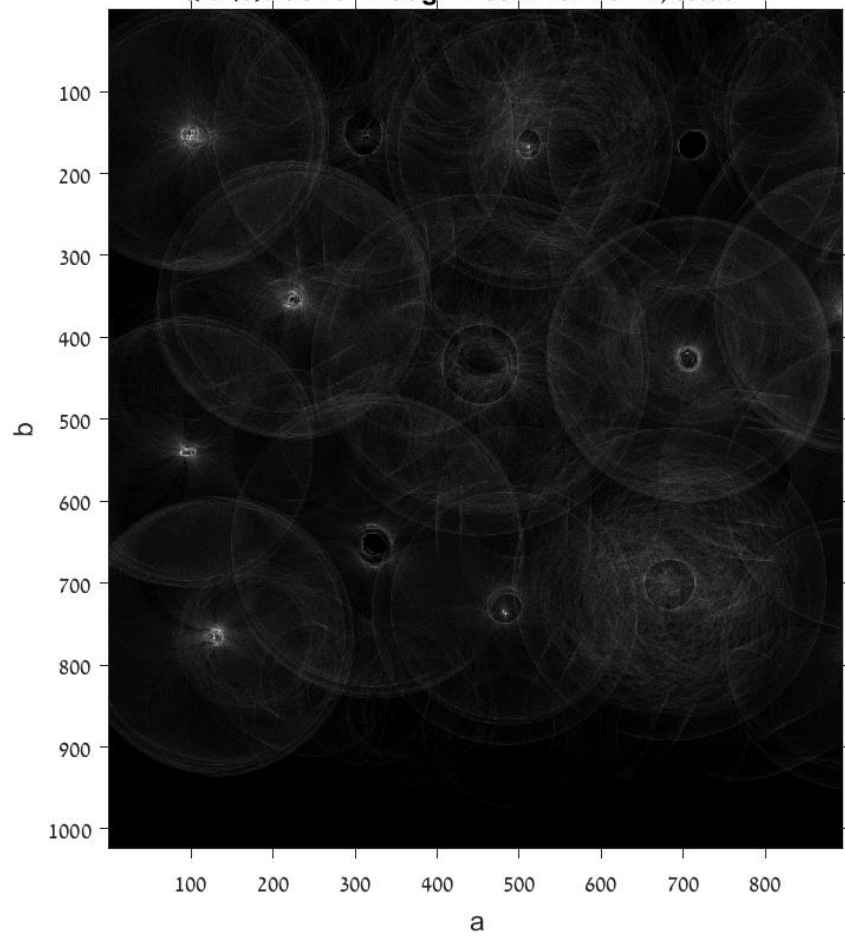
ה-*speedup* שקיבלנו הוא:

$$speedup = \frac{T_{old}}{T_{new}} = \frac{29.12}{6.84} = 4.25$$

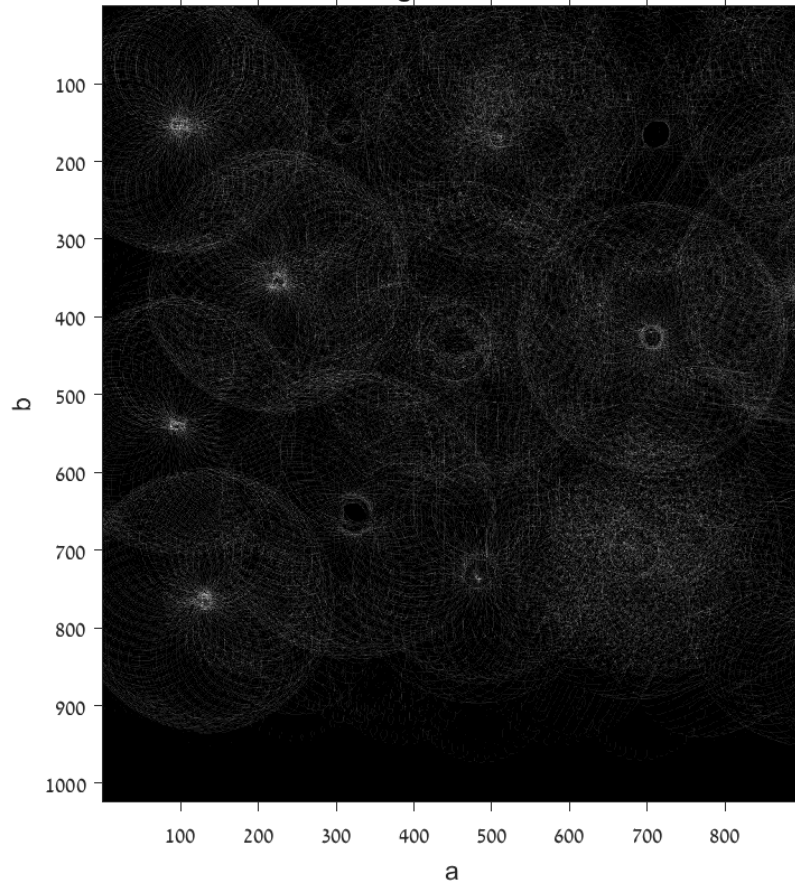
כלומר קיבלנו את אותם הביצועים במהירות פי 4.25 יותר גבוהה.

(e) הצגנו את מטריצת ה-*hough* עבור שתי הקונטיזציות הנתונות ועבור הקונטיזציה שמצאנו בסעיף הקודם. מכיוון שהמטריצה תלת מימדית בחרנו להציג את התמונות עבור  $R=80$  ובכך ליצור תמונה דו מימדית. התוצאות:

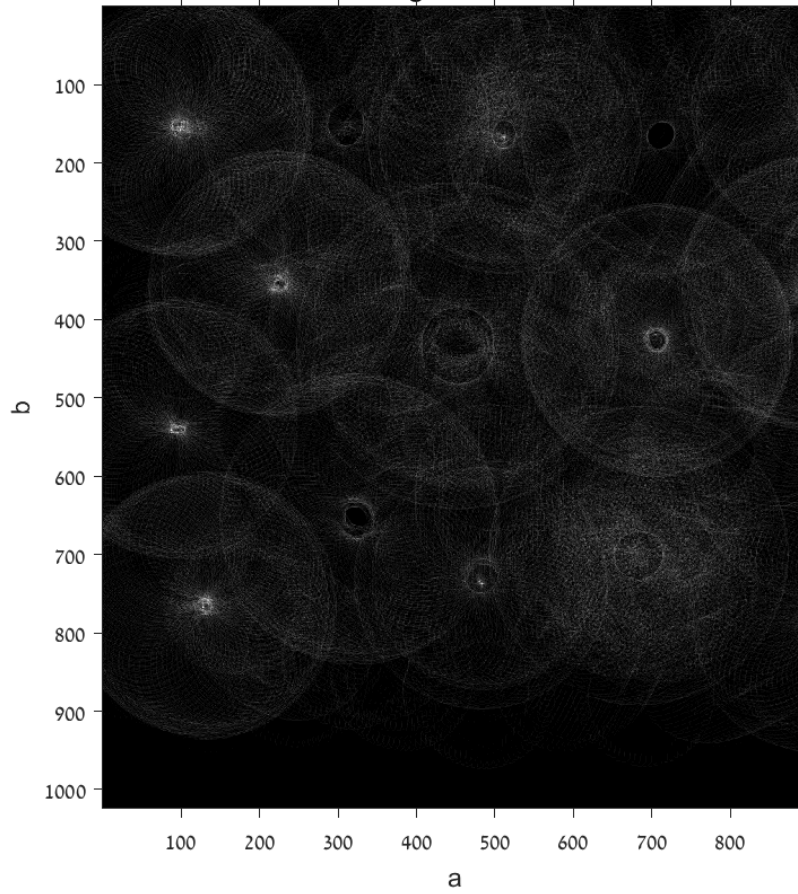
Q2.2(e): Coffee Hough Matrix for  $R_0 = 1$ ,  $\text{teta}_0 = 1$



**Q2.2(e): Coffee Hough Matrix for  $R_0 = 4$   $\text{teta}_0 = 10$**

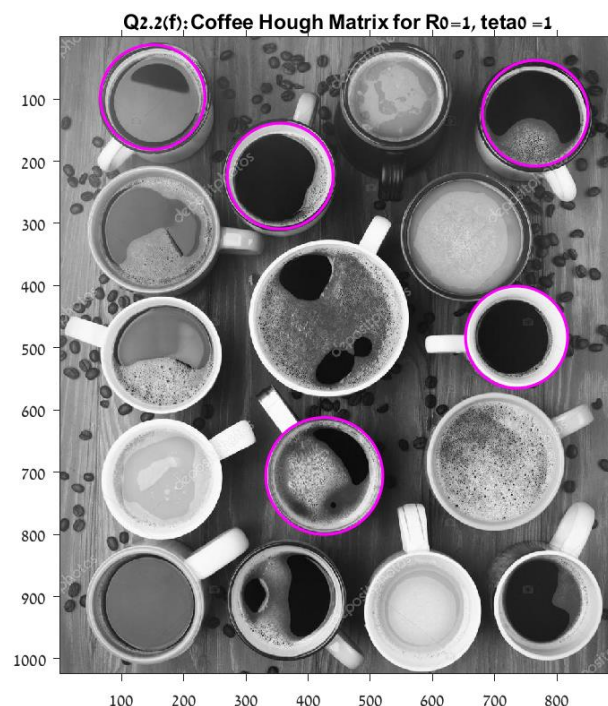


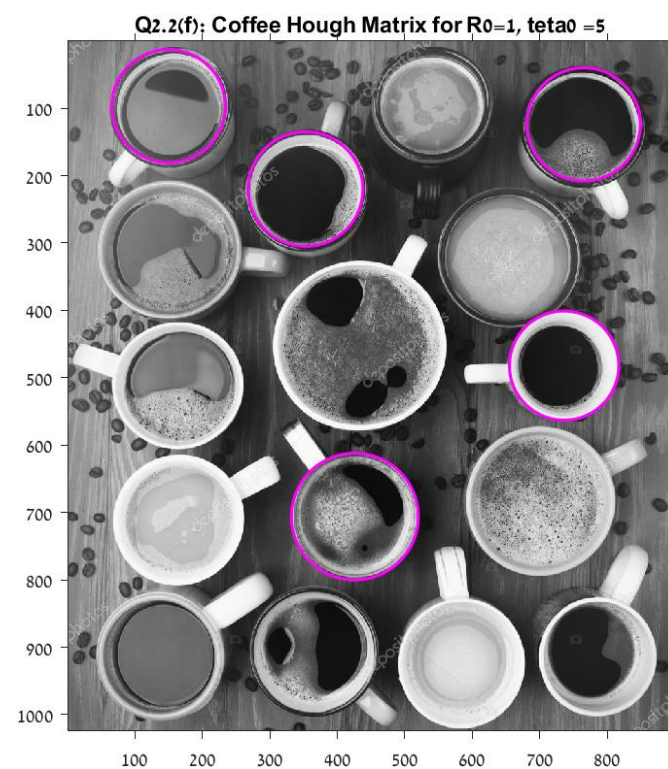
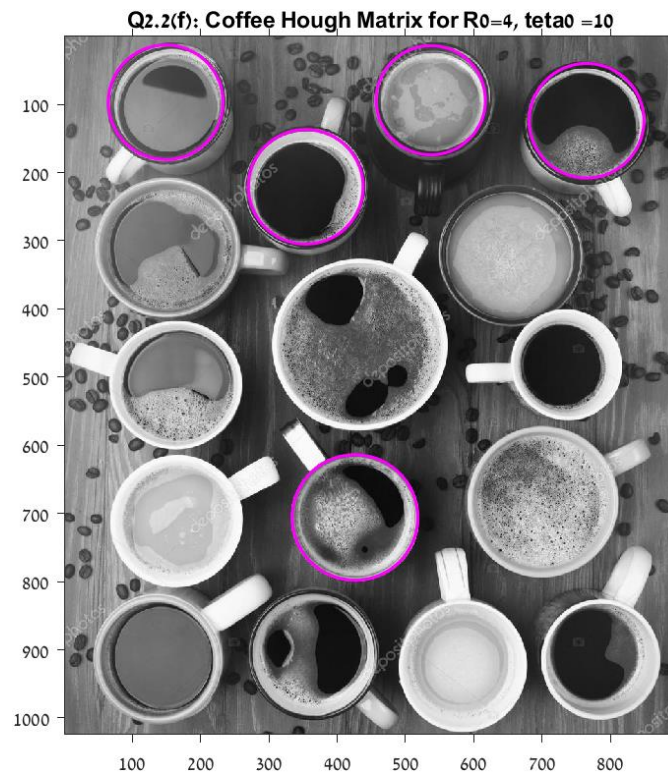
**Q2.2(e): Coffee Hough Matrix for  $R_0 = 1$   $\text{teta}_0 = 5$**



קיבלנו הרבה מעגלים עם מרכזים לבנים חזקים. הנקודות בהם כולם נפגשים הם הנקודות של המעגל המקורי. לכן על מנת למצוא מעגלים דומיננטיים נחפש את נקודות המקסימום במטריצה. ככל שערכי הקוונטיזציה גדולים יותר נקבל פחות דיוק בייצוג המעגל ובעוצמת הלבן בנקודות לו.

(f) כתבנו את הפונקציה *dip\_houghpeaks3d* שמקבלת מטריצת hough ומוצאת את חמשת המעגלים השונים הדומיננטיים ביותר בתמונה. לפי האלגוריתם שנתון בשאלה מצאנו את ערכי הפיק במטריצה ומצאנו את המעגלים הדומיננטיים. ע"מ למצוא מעגלים **שונים** דאגנו למחוק את הערכים במטריצה בסביבת הפיק שכבר מצאנו ברדיוס של 80. (אפשר לראות בתמונה הקצוות שסביב אותו המעגל יש כמה מעגלים לבנים שונים ולכן אנחנו עלולים למצוא את אותו המעגל כמה פעמים בתור מעגל דומיננטי). חישבנו את המעגלים עבור ערכי הקוונטיזציה השונים מהסעיפים הקודמים והצגנו אותם על התמונה המקורית. התוצאות:





נשים לב שעבור  $(R_0, \theta_0) = (1, 1)$  ועבור  $(R_0, \theta_0) = (1, 5)$  קיבלנו את אותם המעגלים הדומיננטים. כלומר, אכן קיבלנו את אותו הדיוק עבור ערכי קוונטיזציה שונים תוך שיפור משמעותי בזמן הריצה.

(g) הסבר על תוצאות הסעיף הקודם : אכן ניתן לראות שקיבלנו עבור כל קוונטיזציה חמישה מעגלים שונים.

עבור ערכי  $R_0$  נמוכים נבדוק רדיוסים שונים רבים יותר (גודל הווקטור  $R$  יהיה 21 לעומת הקוונטיזציה הראשונה עבורה נקבל שגודל הווקטור  $R$  הוא 6), לכן נבדוק הרבה יותר מעגלים ולכן הדיוק שלנו יהיה גבוה יותר, לעומת ערכי  $R_0$  גבוהים עבורם אנחנו עלולים לפספס מעגלים. השימוש בערכי  $R_0$  נמוכים יגרור זמן ריצה גבוה יותר לעומת השימוש בערכי  $R_0$  נמוכים. עבור התוצאות שלנו ניתן לראות שעבור ערכים שונים של  $R_0$  קיבלנו עיגול מודגש שונה אחד, כלומר הספל החמישי שונה בין התמונות. זאת מכיוון שהפונקציה שלנו נותנת פיקים שונים עבור ערכים שונים של  $R_0$  ומכיוון שעברנו על פחות רדיוסים פספסנו את המעגל שמצאנו בקוונטיזציה הראשונה ולכן לא מופיע בקוונטיזציה השנייה.

עבור ערכי  $\theta_0$  נמוכים נקבל מעגלים ברורים וחלקים יותר כי נעבור על יותר זוויות של המעגל עצמו. אם נגדיל את  $\theta_0$  נקבל מעגלים פחות ברורים והערכים שלהם במטריצת ה-hough יהיו נמוכים יותר, לכן אנחנו עלולים לפספס נקודות על העגלים ולקבל פיקים שונים ומכך גם לפספס את המעגלים הדומיננטיים. גם במקרה הזה היתרו שבערכי  $\theta_0$  גבוהים יותר הוא זמן ריצה קצר יותר.

בכל מקרה ראינו שעבור שינוי בלבד בערכי  $\theta_0$  נקבל זמן ריצה הרבה יותר קצר ודיוק זהה במציאת הפיקים.