

## תרגיל בית 1 ראייה ממוחשבת:

### שאלה 1

הרעיון שעומד מאחורי פתרון השאלה הוא ביצוע השלבים הבאים:

- יצירת Edge Map כדי שנוכל להשתמש בנקודות שנמצאות על האליפסות שבתמונות.
- מציאת מרכזי האליפסות באמצעות לקיחת שתי נקודות על אליפסה כלשהי ושימוש במשוואה שנתונה במאמר, והוספת "הצבעות" לכל נקודה שתהיה מועמדת להיות מרכז מעגל
- בחירת הנקודות שקיבלו הכי הרבה הצבעות להיות מרכזי האליפסות שבתמונה, ושמירת הנקודות שהצביעו לאותה נקודת מרכז אליפסה
- מציאת  $B, C, D$  מהמשוואה הנתונה במאמר ע"י כך, שעבור כל נקודת מרכז של אליפסה שמצאנו, ועבור כל  $D$  ו- $B$ , נשתמש בנקודות שהצביעו לאותה נקודת מרכז ע"י הצבתן במשוואה הנתונה (וכך נמצא את  $C$ ), והוספת הצבעות לאותם  $B, C, D$  באמצעות קובייה תלת מימדית בגודל  $9 \times 9 \times 9$ , שבכל איטרציה, תצמצם את טווחי  $B, C, D$  שגבולותיה מסמלים, עד להתכנסות של ערכים מדויקים.
- בחירת התא בעל מספר ההצבעות המקסימלי בקובייה
- הצבת ערכי  $B, C, D$  במשוואות מהמאמר על מנת למצוא את ערכי  $a, b, \theta$
- ציור האליפסות שהתקבלו על התמונה הנתונה

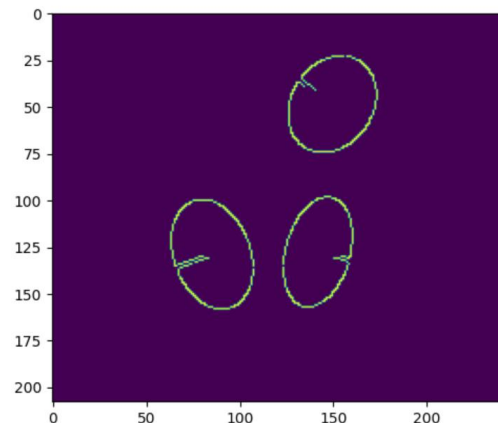
האופן שבו עובד ה-script שמציג את התמונות הוא שהוא נותן לכל תמונה מרחק מקסימלי ומינימלי בין שתי נקודות על אליפסה כך שאם נבחרו שתי נקודות בתמונה, שהמרחק שלהן גדול או קטן מהמרחקים הנתונים, שתי הנקודות לא נכנסות לחישוב מרכז האליפסה. דבר זה עוזר להימנע מסימון של אליפסות שאינן בתמונה וחוסך זמן ריצה רב.

בנוסף ניתנים הפרמטרים הבאים:

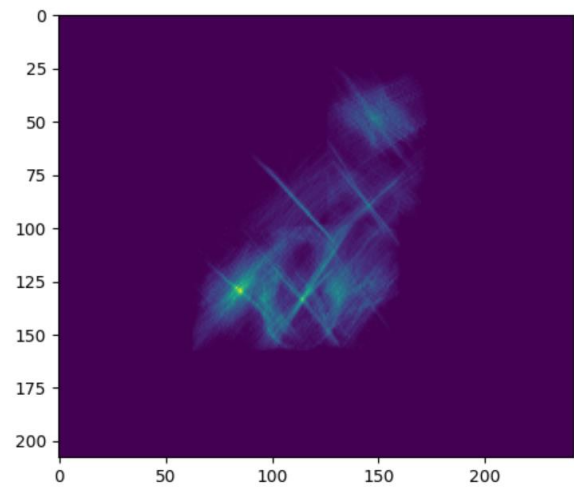
- פרמטרים אשר עוזרים ליצירת ה-Edge map, האמצעות טשטוש גאוסייני והפונקציה `cv2.Canny`
- גודל רדיוס הנקודה שתסמן את מרכז האליפסה שהאלגוריתם מצא
- פרמטרים לנרמול גודל האליפסה שהאלגוריתם מצייר על התמונה הנתונה
- מספר הנקודות הרנדומליות שיש להשתמש בהן על מנת למצוא תא מרכז המעגל בהתאם לאופן המתואר במאמר הנתון
- טווח חיפוש נקודות בעלות מקסימם מקומי של מספר הצבעות. למשל עבור הטווח  $[100, 100]$  האלגוריתם בקוביות של  $100 \times 100$  נקודות בעלות מספר הצבעות מקסימלי.

### תמונה מספר 1:

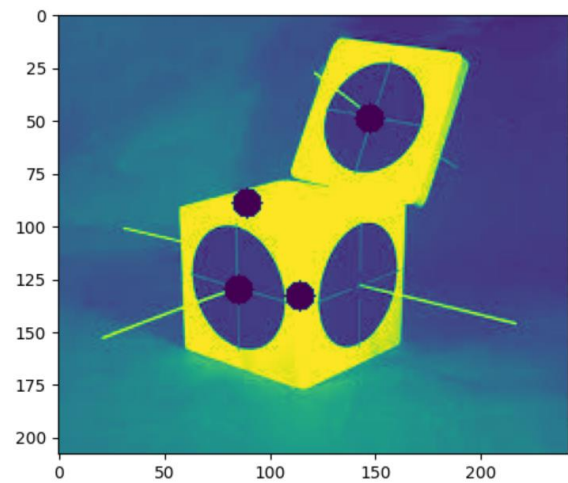
ניצור Edge Map באמצעות `cv2.Canny`



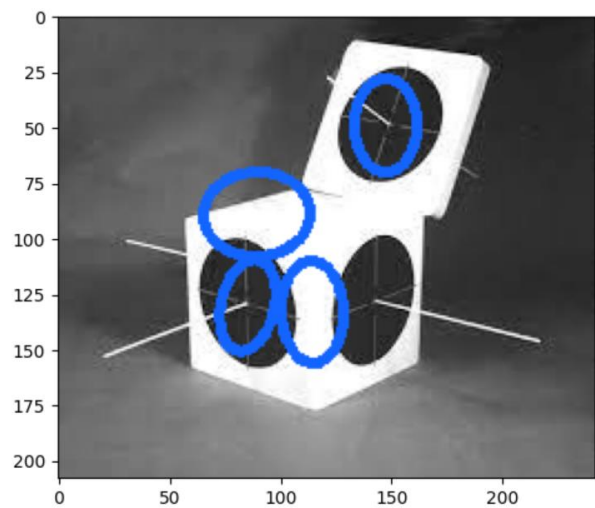
ניתן לראות את מפת ההצבעות עבור התמונה הנתונה (הפיקסלים בעלי הערך הצהוב ביותר קיבלו הכי הרבה הצבעות):



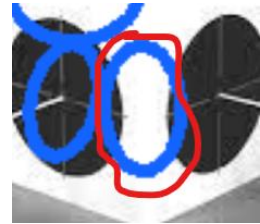
נעבור על כל הפיקסלים בתמונה ונבחר את הנקודות המקומיות בעלות הערכים המקסימליים:



ומכאן, ע"י מעבר על כל ה-d, b, והנקודות שמצביעות אל אותם מרכזים, נקבל את האליפסות הבאות:

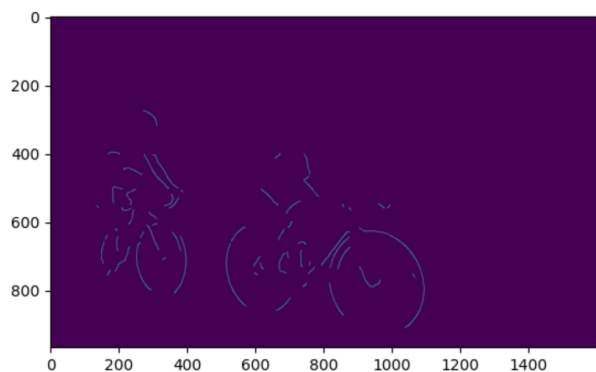


ניתן לראות שמצאנו שני מרכזים של אליפסות מתוך שלוש. ניתן להסיק ששתי האליפסות השגויות שמצאנו זוהו ע"י האלגוריתם בגלל הקירבה של האליפסות אחת לשניה. כיוון שבשביל למצוא תא מרכז האליפסה, אנחנו בוחרים בכל פעם שתי נקודות רנדומליות מה-Edge map שיצרנו, סביר מאוד שנבחרו שתי נקודות מאליפסות שונות. ולכן ניתן להסיק שלמשל, האליפסה הבאה:



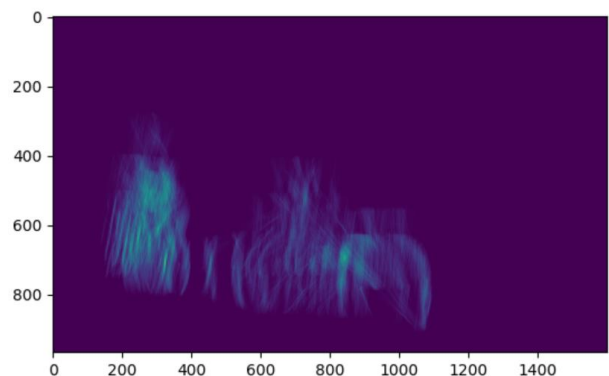
נוצרה בגלל שהמרכז שלה קיבל הרבה הצבעות מנקודות שנמצאות על האליפסות שמשני הצדדים שלה. תופעה זו קרתה בכל אחת מהתמונות ואין דרך להימנע ממנה, מכיוון שהמרחק בין האליפסות יכול להיות קטן יותר ממרחק של שתי נקודות על אותה אליפסה, ולכן אם רוצים למצוא אליפסות כלשהן, בליט ברירה, יופיעו גם מרכזים של אליפסות שאינן קיימות בתמונה.

תמונה מספר 2:

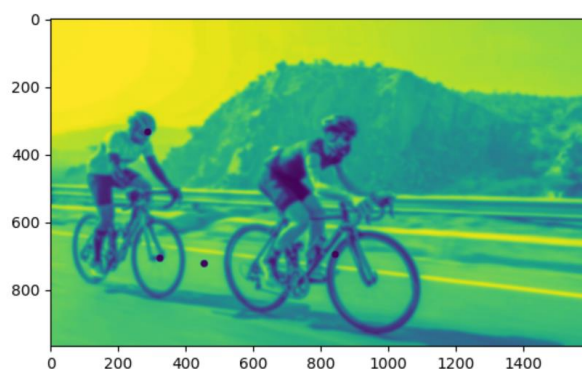


ניתן לראות שעל אף השימוש בטשטוש גאוסייני ובשיטת Canny בשביל ליצור Edge map נקי ככל האפשר, עדיין קיבלנו נקודות שלא נמצאות על האליפסות. הנקודות האלו יפריעו להתנהלות התקינה של האלגוריתם על כך שהן יוסיפו הצבעות למרכזים של אליפסות שלא קיימת בתמונה, ולכן הדבר יקשה על מציאת מרכזי האליפסות האמיתיות שנמצאות בתמונה.

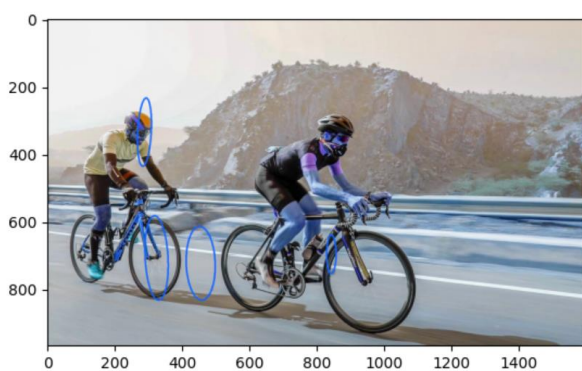
מפת ההצבעות המתקבלת מה-Edge map שיצרנו היא:



ומכאן, המרכזים המקסימליים שהאלגוריתם יסמן הם:

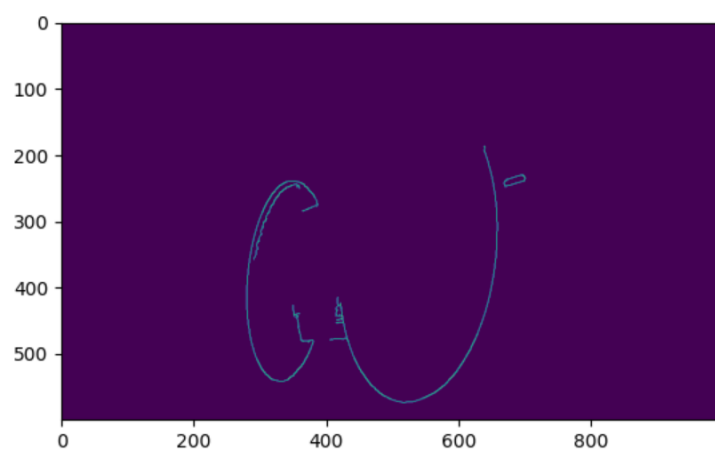


ניתן לראות שהצלחנו למצוא את מרכז האליפסה משמאל למטה (הגלגל הקדמי של הרוכב השמאלי), ואת מרכז האליפסה שנוצרת מהקסדה של הרוכב השמאלי, אך כאמור, מצאנו גם מרכזים שלא קיימים בתמונה ולכן נקבל את התוצאה הבאה:

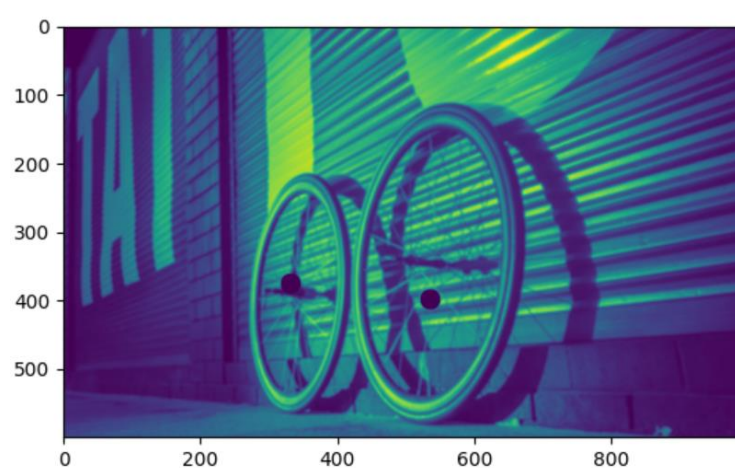
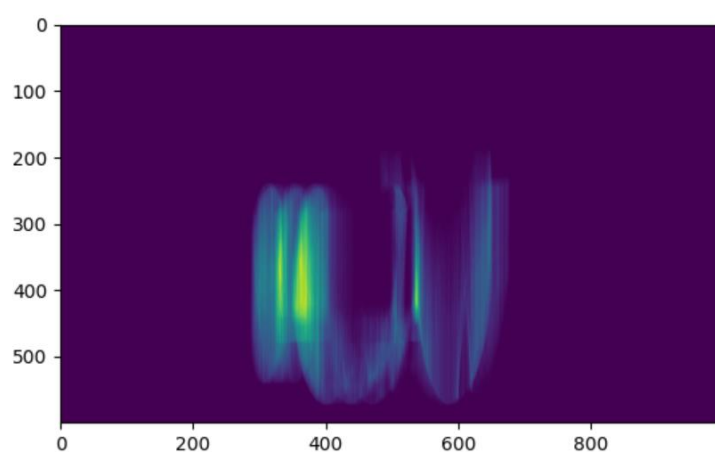


תמונה מספר 3:

ניצור Edge map:



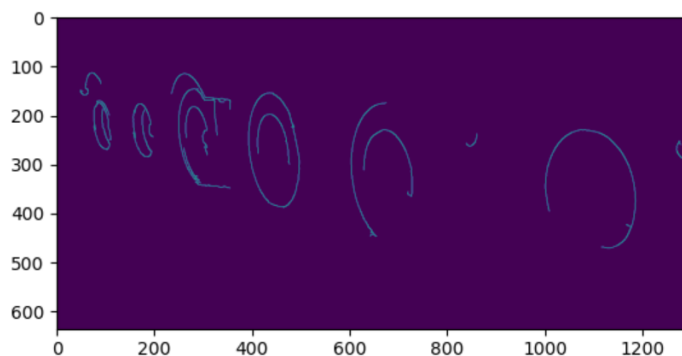
נקבל את מפת ההצבעות הבאה:



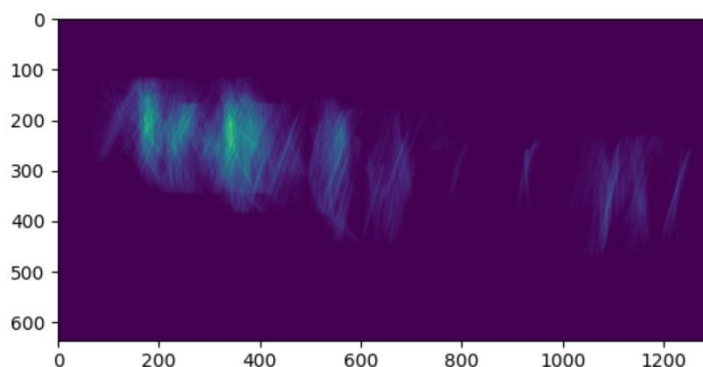


#### תמונה 4

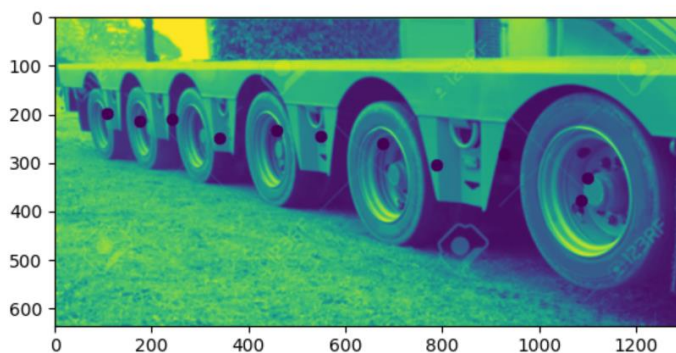
ניצור Edge map:



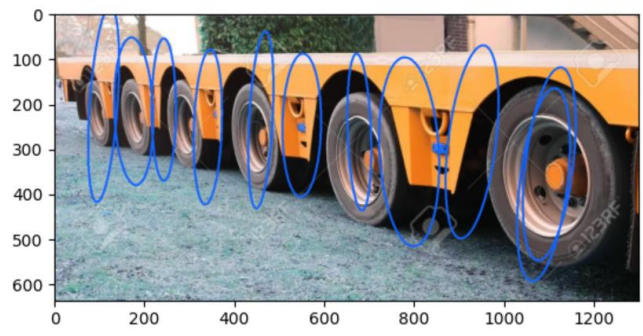
כיוון שחלק מהאליפסות קרובות אחת לשנייה, כמו בתמונה 1, נקבל הצבעות לאליפסות שלא קיימות בתמונה. מכאן, מפת ההצבעות תראה כך:



והנקודות מעלות מספר הצבעות מקומי מקסימלי יהיו:



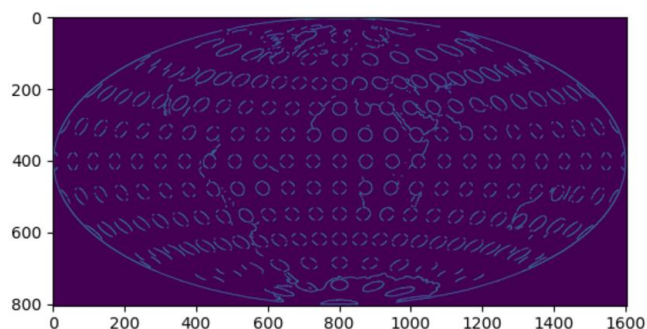
ניתן לראות שבגלגלים שיותר רחוקים אחד מהשני, האלגוריתם מצליח למצוא מרכזים של אליפסות שקיימות בתמונה (הגלגל, ומסגרת המתכת בצורה של חצי אליפסה מעל הגלגל) אך בגלגל שקרובים יותר אחד לשני, אנחנו מקבלים נקודות פחות מדויקות באופן יחסי. מכאן, האליפסות שנקבל יהיו:



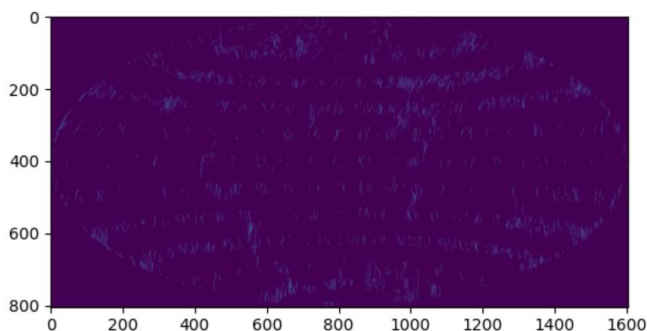


## תמונה 5

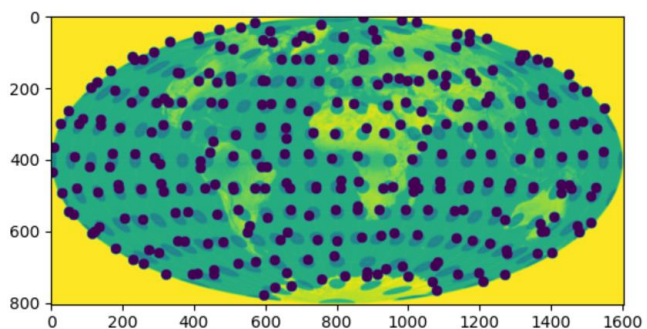
ניצור Edge map:



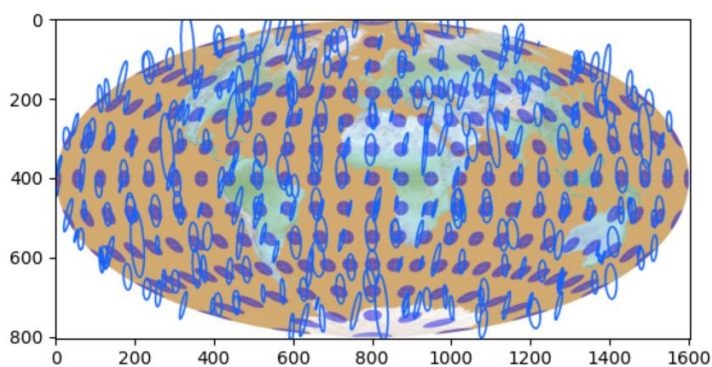
בתמונה זו, כיוון שקיימות הרבה אליפסות קטנות, נחפש אליפסות שהנקודות שיהיו עליהן יהיו במרחק קטן אחת מהשנייה. מכאן, מפת ההצבעות שנקבל תהיה:



והנקודות על מרכזי האליפסות יהיו:

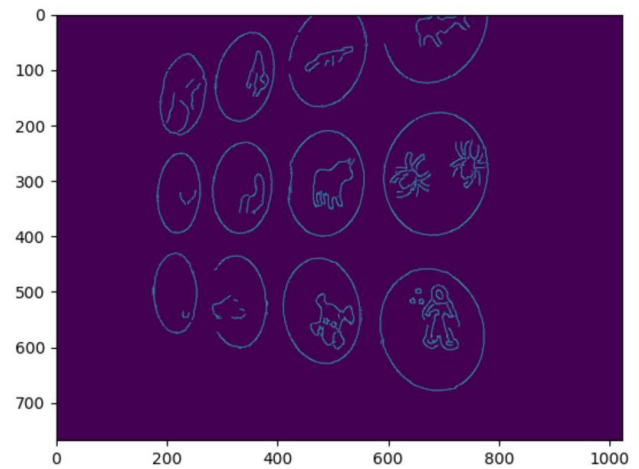


והאליפסות עצמן:



## תמונה 6

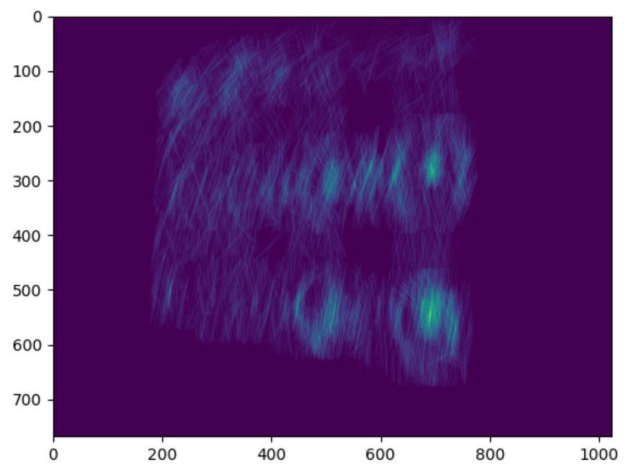
ניצור Edge map:



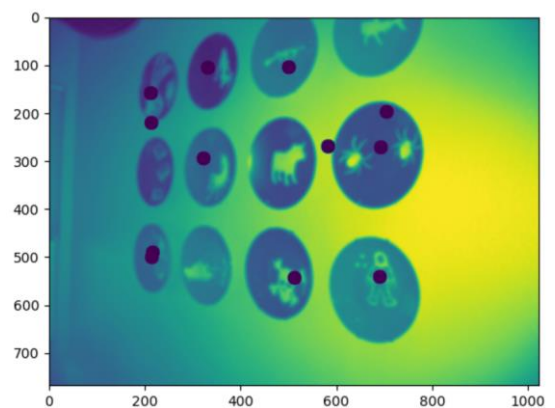
כיוון שיש ציורים קטנים בתוך האליפסות, שטשטוש גאוסייני ו-Canny לא מצליחים להוציא מתמונה, מתווספות נקודות רבות שאינן נמצאות על אליפסות בתמונה ולכן נקבל הצבעות לאליפסות שאינן נמצאות בתמונה.

בשביל לצמצם את הנזק שהציורים הקטנים יצרו, נשתמש רק בנקודות שבאופן יחסי רחוקות אחת מהשנייה.

מכאן, מפת ההצבעות תהיה:

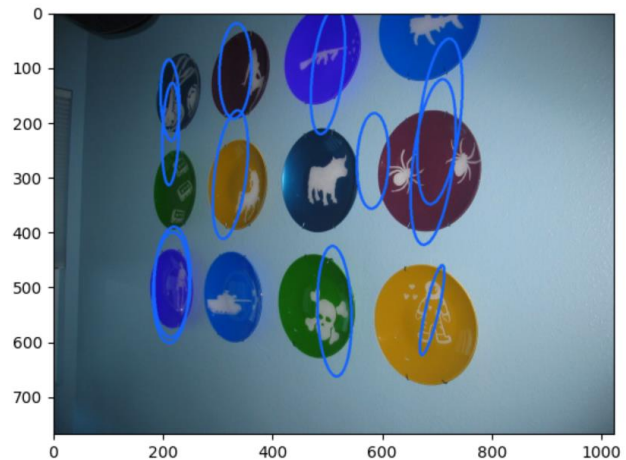


ומרכזי האליפסות יהיו:



ניתן לראות שכן שהצלחנו למצוא חלק ממרכזי האליפסות אך חלק מהמרכזים השתבשו בגלל הצבעות לא נכונות.

מכאן, האליפסות שנקבל יהיו:



**אופן מימוש ציור האליפסות:**

אנחנו מאתחלים מילונים שמכילים טווח אפשרי ל-D, C, B מהמשוואה שבמאמר.

אנחנו עוברים על כל אחד מהמרכזים שמצאנו, ומשתמשים בכל הנקודות שהוסיפו הצבעות לאותו מרכז עליו אנחנו עוברים. בכל איטרציה של הלולאות הפנימיות אנחנו מציבים את כל אחת מהנקודות שהצביעו למרכז המעגל, במשוואה הנתונה במאמר, וסופרים הצבעות לכל אחד מהערכים האפשריים של B ו-D בטווח המוגדר של המילונים שלהם. עבור כל נקודה אנחנו מקבלים הצבעה במטריצה בגודל 9X9X9, ואנחנו עוקבים אחרי התא בעל מספר ההצבעות המקסימלי.

אחרי שעברנו על כל הערכים האפשריים של B ו-D, וחישבנו איתם C-ים שונים, אנחנו מעדכנים את הטווחים החדשים של המילונים באמצעות התא שקיבל מספר הצבעות מקסימלי.

הטווח החדש יחושב באופן הבא.

אם תא מסויים מתאר את הטווח 100 עד 200, הטווח החדש של המילון כולו יהיו בין 100 ל-200. ונעשה זאת עבור כל אחד מהמילונים.

נמשיך באופן הזה עד שלבסוף כל אחד מהמילונים ייתכנס לערך קבוע כלשהו ונקבל את האליפסות המתוארות באמצעות D, C, B שמצאנו.

את D, C, B שמצאנו נציב במשוואות הנוספות שנתונות במאמר, וכך נקבל את a, b, theta שבאמצעותן נצייר את האליפסות על התמונה.

