תוכן עניינים

מבוא

ניווט ניתן לתיאור כתהליך למציאת מסלול מתאים ובטוח לנסיעה ממקור ליעד.

לרובוטים ניידים יש חשיבות רבה בתעשייה, תחבורה, מערכות ביטחוניות וכו'.
לרובוטים אלו דרושה היכולת לנוע בסביבה מורכבת ולא ידועה. בדרך כלל יכולת זו תסתמך על מערכות חישה ואלגורתמי ניווט.

שיטות קיימות

כיום קיימות שתי שיטות מרכזיות לניווט :

1. ניווט בעזרת מפה –
שיטה זו מבוססת על כך שהרובוט מקבל מודל של הסביבה בה הוא ינווט ,כאשר למודל רמת פירוט התלויה בהגדרת המחקר.
שיטה לבנית המודל של הסביבה היא היטל דו מימדי של הסביבה המכיל פירוט של כל נקודות העניין בסביבה שבעזרתן הרובוט ינווט.
שיטות מתקדמות יותר הגדירו לכל מכשול כוח דוחה כלפי הרובוט ובכך הרובוט ימנע מלסוע לכיוונם.
האלגוריתם הכללי למערכות בשיטה זו :
   a. רכישת תמונה
   b. זיהוי נקודות עניין ומאפיינים בתמונה
   c. התאמת מאפיינים שזוהו למודל מפה של הרובוט
   d. עדכון מיקום הרובוט כפונקציה של מצייני המיקום שסומנו כמזוהים במפה
2. ניווט ללא מפה – מבוסס על שיטות תגובה אשר משתמשות ברמזים ויזואלים הנגזרים מסגמנטצית התמונה.

לפתירת בעית מיקום של הרובוט יש שתי גישות :

1. מיקום אבסולוטי – המיקום ההתחלתי של הרובוט לא ידוע . מיקום הרובוט נקבע ע"י שיטות של מציאת מאפיינים ושינויים במיקומי מאפיינים שונים אשר נשארים אינוריאנטים לנקודת מבט של הרובוט.
2. מיקום רלטיבי – מבוסס על ההנחה שבתחילת הניווט מיקומו של הרובוט בעולם ידוע.

ניתן לחלק את בעית תיכנון המסלול לשתי גישות :

1. תכנון גלובלי – קיים ידע מוקדם על סביבת הנסיעה ונבנה מסלול מהמקור ליעד כאשר הוא מתעלם ממכשולים אפשריים בדרך.
2. תכנון לוקלי – בכל שלב בנסיעה מתוכנן המסלול הלוקלי לנסיעה על סמך המכשולים שזוהו.

הפרויקט הנוכחי תוכנן בשיטות של ניווט בעזרת מפה , מיקום הרובוט הוא רלטיבי,ידוע מראש ותכנון המסלול הוא גלובלי אך כאשר מזוהה מכשול תהליך הניווט נעצר עד שהמכשול מוסר מדרכו של הרובוט.

רקע לפרויקט

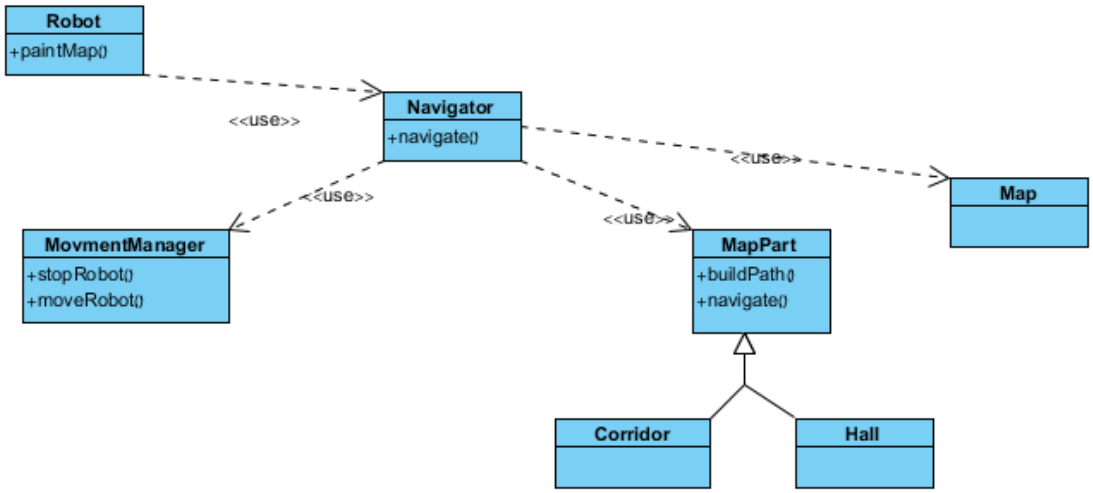
יעדים

בניית מערכת ראייה על גבי פלטפורמת רובוט,כאשר יכולות המערכת הן :

1. ניווט במסדרונות ממקור ליעד (במונחי דלתות) תוך שימוש במפה ומצלמת קינקט (צבע ועומק) בלבד.
2. יעדי משנה :
   a. זיהוי מסדרון – זיהוי כניסה/יציאה ממסדרון.
   b. נסיעה במרכז המסדרון.
   c. זיהוי דלתות.
   d. עדכון מיקום הרובוט במפה על סמך סימנים מוגדרים שיזהו ע"י מערכת הראיה (דלתות).
   e. זיהוי מכשולים בדרכו של הרובוט בנסיעתו במסדרון.

סקירה כללית

מבנה המערכת



1. MapPart – מודל המייצג רכיב בקומה. בונה את המסלול לפי מקור ויעד ומבצע ניווט בחלק זה על פי תמונות צבע ועומק הניתנות לו. קיימים שני חלקים הנתמכים במערכת :
   a. Corridor – מנהל נסיעה באזור מסדרון.
   b. Hall – מנהל נסיעה באזור של חלל כניסה לקומה.
2. Map – מודל המנהל את מיקום הרובוט במפה, ובונה את המסלול כרשימה של MapPart.
3. MovmentManager – מודל המנהל את פקודות הנסיעה של הרובוט. מקבל כיוון (ימינה, שמאלה או ישר) ופקטור 0 עד 1 הקובע את חדות הפניה.
4. Navigator – מודל המנהל את הניווט.
5. Robot – מודל ההרצה המקבל את שירותי הניווט ממודל Navigator.

<u>זרימת המידע במערכת :</u>

1. טעינת מפה למודל ה- Map, ואתחול חדר מקור וחדר יעד עבור ה- Navigator .
2. Robot דוגם מהמצלמה תמונת עומק ותמונת צבע. מבצע קריאה ל- Navigator.navigate עם התמונות.
3. Navigator מבצע קריאה ל- MapPart.navigate הנוכחי במסלול.
   a. אם ערך ההחזר הוא CONTINUE – נותן פקודת נסיעה לרובוט.
   b. אם ערך ההחזר הוא FOUND_OBSTACLE – נותן פקודה לרובוט לעצור.
   c. אם ערך ההחזר הוא EXIT_PART – מתקדם לרכיב הבא במסלול.
   d. אם ערך ההחזר הוא FOUND – עוצר ליד הדלת יעד.
4. ה- Robot לאחר פקודת הניווט לוקח את רכיב ה- Map ומעדכן את התצוגה על פיו.

Corridor.navigate:

1. מציאת גבולות המסדרון.
    a. אם לא נמצא – החזר OBSTACLE_FOUND.
2. חיפוש דלת הכי קרובה מימין למסדרון ודלת הכי קרובה משמאל למסדרון.
    a. קביעה אם אלו דלתות חדשות.
        i. אם כן עדכן את רכיב המפה.
            1. אם זו דלת היעד החזר FOUND.
3. בדיקה אם נכנסים לחלק של חלל כניסה לקומה.
        i. אם כן החזר EXIT_PART.
4. מרכוז הרובוט במסדרון - קביעת כיוון הנסיעה של הרובוט .
5. החזר CONTINUE.

Hall.navigate:

1. חיפוש דלת הכניסה למסדרון.
    a. אם נמצא – החזר EXIT_PART.
2. חישוב כיוון הפניה ועד כמה חדה תהיה.
3. החזר CONTINUE.

פירוט נרחב על כל שלבי אלגוריתמי הניווט של Corridor ו Hall יוצג בסעיפים הבאים.

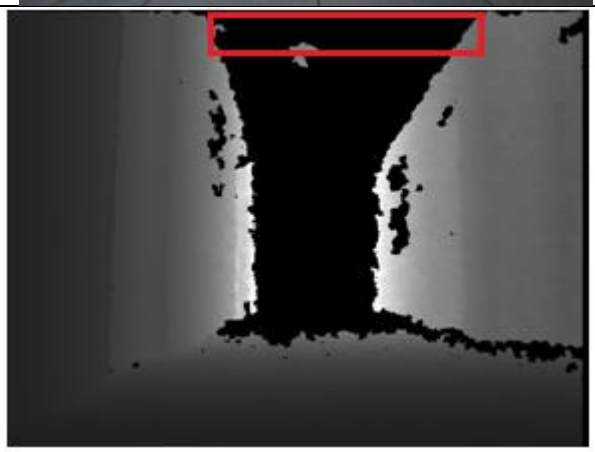
פירוט מודלים עיקריים

נסיעה במסדרון

נסיעה במרכז המסדרון

על מנת שהרובוט לא יתקל בקירות ויזהה דלתות על פי הסדר שהן ממוקמות על הרובוט לנסוע במרכז המסדרון ולהתאים את עצמו לשינויים של התרחבות והצרות של המסדרון.

לדוגמא :
הדלת מימין ממוקמת אחרי הדלת משמאל.

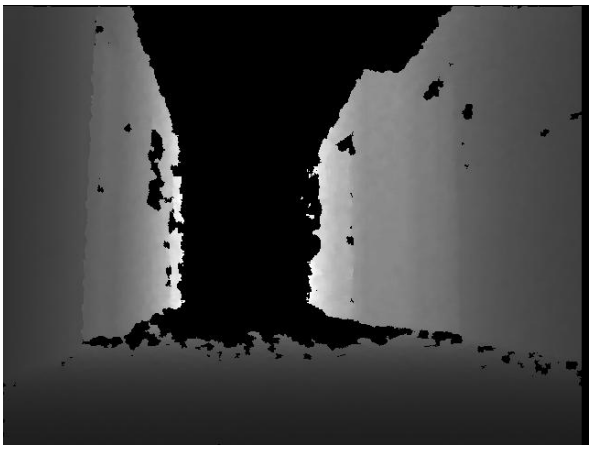
רובוט לא במרכז	רובוט במרכז
	
	

ניתן לשים לב כי החלק העליון המסומן בתמונת העומק מעיד על חוסר המרכוז של הרובוט בתמונה הימינית לעומת התמונה השמאלית.

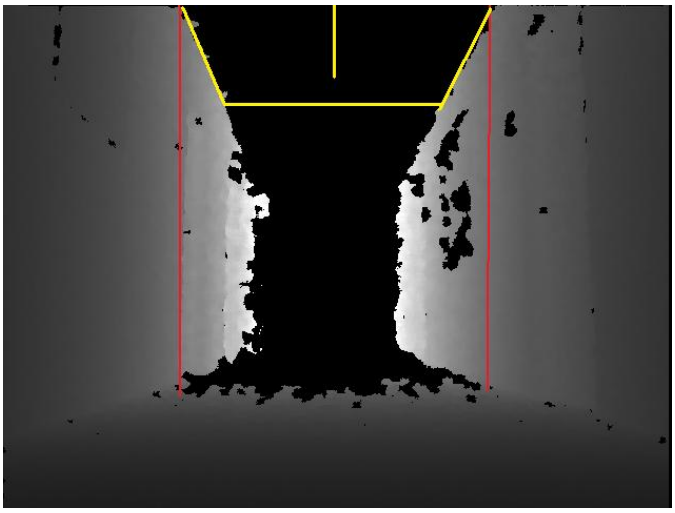
החלק העליון השחור כפי שמוצג בתמונות העומק לעיל הוא התקרה של המסדרון וכיוון שהוא קרוב לרובוט, ייתן חיווי של התרחבות או היצרות של המסדרון באופן מידי. זאת בניגוד למלבן השחור שבמרכז תמונת העומק המייצג את המשך המסדרון ויבטא את רוחב המסדרון הרחוק יותר.

דוגמא להתרחבות המסדרון :

דוגמא להיצרות המסדרון :



על סמך מאפיין זה, נגדיר את מרכז המסדרון כאמצע הקטע השחור העליון בתמונת העומק (מסומן בצהוב) :



מציאת מרכז המסדרון

1. רוץ על שורות 20 עד 70 בתמונת העומק
   a. רוץ על עמודות השורה הנוכחית
      i. עמודת הפיקסל השחור הראשון מהווה את החלק השמאלי של המסדרון
      ii. ספור פיקסלים שחורים בשורה
      iii. אם הפיקסל הנוכחי אינו שחור
         1. אם מספר הפיקסלים השחורים גדול מ 180
            a. העמודה הנוכחית מהווה את החלק הימיני של המסדרון
         2. אחרת אפס מונה
2. לאחר קבלת העמודות של החלקים הימיני והשמאלי של המסדרון חשב את האמצע המסדרון.

לאחר שקיבלנו את אמצע המסדרון עלינו להחליט אם לפנות ימינה ,שמאלה או להמשיך ישר. באופן אידיאלי, כאשר הרובוט יהיה במרכז המסדרון, אמצע המסדרון שחושב יהיה בעמודת אמצע התמונה (עמודה 320). לכן נחליט את כיוון הפניה כך :

חישוב כיוון הפניה

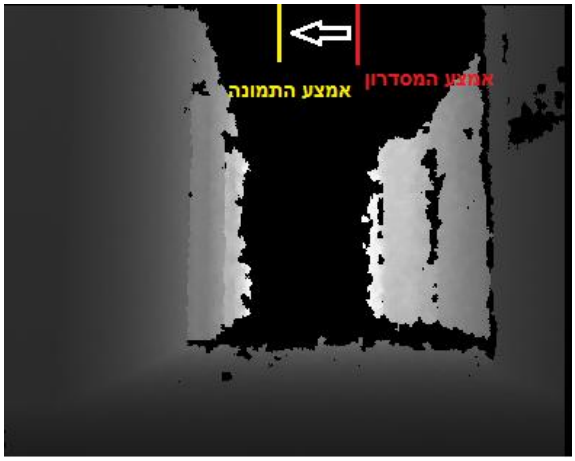
1. בדוק אם אמצע המסדרון שחושב הוא בסביבת עמודת אמצע התמונה (5 פיקסלים ימינה ושמאלה).
2. אם כן המשך ישר
3. אחרת
   a. חשב את הפרש אמצע המסדרון מאמצע התמונה (בערך מוחלט) וחלק ב-315. נסמן ביטוי זה בתור factor.
   b. אם אמצע המסדרון משמאל לאמצע התמונה סע שמאלה.
   c. אם אמצע המסדרון מימין לאמצע התמונה סע ימינה.

לאחר שהוחלט על כיוון הפניה (במידת הצורך) עלינו לדעת עד כמה חדה הפניה. במצב בו אנו קרובים מאוד לאחד מקירות המסדרון עלינו לבצע פניה חדה ואם אנו באמצע המסדרון אך הוא הוצר או התרחב מעט עלינו לבצע פנייה עדינה. זאת נעשה בעזרת הfactor ככל שהוא יהיה גדול יותר כך הפניה תהיה חדה יותר.

דוגמא – קירבה לקיר ימין של המסדרון – עלינו לבצע פניה חדה שמאלה :

ניתן לראות כי אמצע המסדרון הוא משמאל לאמצע התמונה (המייצגת את אמצע המסדרון הרצוי) ולכן הfactor יהיה גדול ולכן הפניה תהיה חדה.

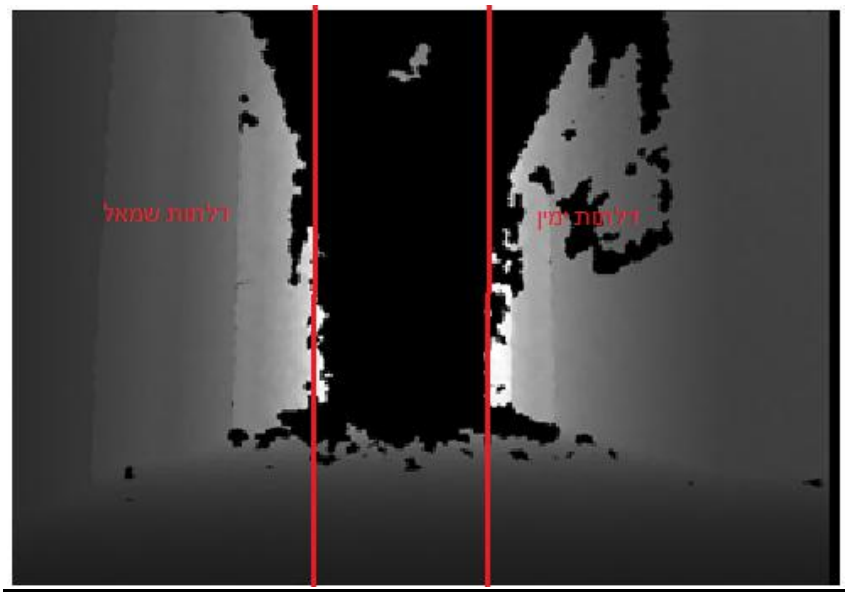
דוגמא – פניה עדינה לתיקון מרכוז הרובוט במסדרון – עלינו לבצע פניה ימינה :



במקרה זה ההפרש יהיה קטן ולכן גם הfactor ולכן הפניה תהיה עדינה.

זיהוי דלתות

על מנת למצוא את דלתות המסדרון עלינו להגדיר את החלק הימיני והחלק השמאלי של המסדרון.
כפי שניתן לראות בתמונות עומק שהוצגו לעיל במסמך קל להבחין עי המלבן השחור באמצע המסדרון מגדיר את האופק של המסדרון והחלק האפור מימינו מכיל את דלתות ימין של המסדרון וכך עבור החלק השמאלי.

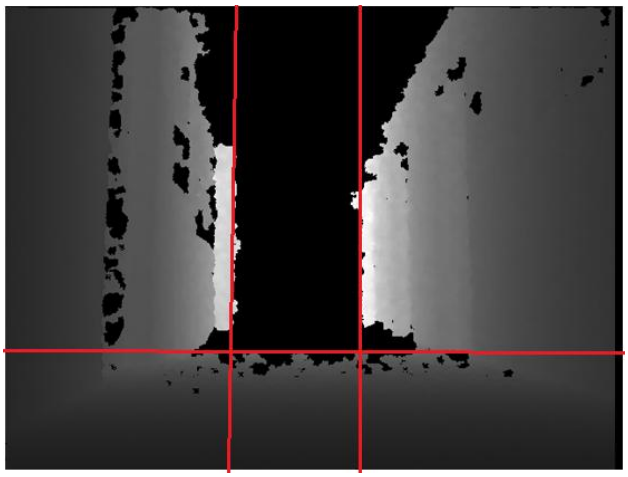


<u>מציאת חלק ימין ושמאל של המסדרון</u>

1. סכימת עמודות משורה 120 עד 370 (אלו שורות שברוב התמונות המלבן השחור המייצג את אופק המסדרון מופיע)
2. סריקת וקטור סכום העמודות משמאל לימין
   a. עבור התא הראשון המכיל מעל 210 פיקסלים שחורים נסמן אותו כגבול שמאל של המסדרון
      i. ספור כמות תאים המכילים מעל 210 פיקסלים
      ii. עבור התא הראשון המכיל פחות,בדוק את הרוחב שהתקבל, אם מעל 60 סמן אותו כגבול ימין של המסדרון ועצור
      iii. אחרת התחל מחדש לחפש גבול שמאלי של המסדרון

כפי שניתן לראות אלגוריתם זה מחפש מלבן שחור ברוחב 60 וגובה 210 ומתעלם ממלבנים קטנים ממנו ובכך אנו מתעלמים ממצבים של דלתות פתוחות היכולות להטעות,לדוגמא בתמונה הבא אנו מתעלמים מהמלבן השחור הצר בצד שמאל של התמונה המהווה דלת פתוחה.

לאחר שהתקבלו גבולות המסדרון נחפש דלתות בחלק השמאלי ובחלק הימיני ע"י כך שנגזור מתמונת הצבע החלק השמאלי הוא הגבול השמאלי של המסדרון עד קצה שמאל של התמונה והחלק הימיני של המסדרון הוא הגבול הימיני של המסדרון עד הקצה הימיני של התמונה.בנוסף לכך נתעלם מהרצפה ונקבל :



נגזור את תמונת הצבע על פי גבולות הנ"ל ונקבל :

לאחר שיש לנו כל חלק נותר השלב של זיהוי הדלתות בכל חלק. ניתן לשים לב להבדל הצבעים החד בין הדלת לקיר ולמבנה של הדלת : גבוהה וכהה. נרצה לזהות את החלק הרחוק של הדלת על מנת שתהיה כמות מספקת של זיהויי אותה דלת.על סמך כל מאפיינים ודרישות אלו נבנה פילטר המחפש מעבר חד מכהה לבהיר בכיוון אלכסוני.

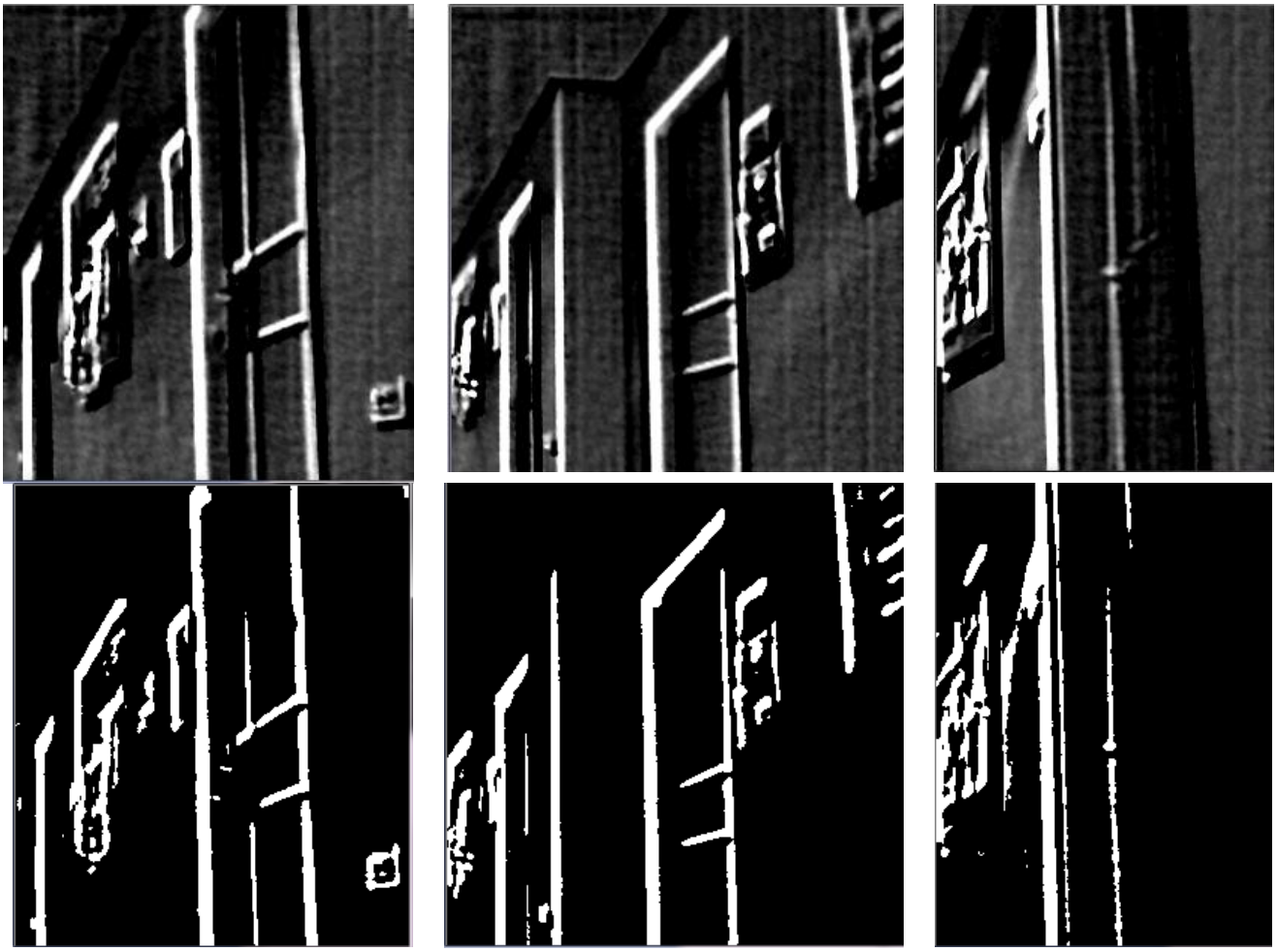
דוגמא לפילטר עבור החלק הימיני של המסדרון :

3	1.5	0	0	0	0	0
1.5	2	0.5	0	0	0	0
0	0.5	1	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	-1	-0.5	0
0	0	0	0	-0.5	-2	-1.5
0	0	0	0	0	-1.5	-3

ניתן לראות את ההעדפה לאלכסון. ניתן לראות שאם נעמוד על פיקסל שהוא המשקוף השמאלי של דלת בצד הימיני של המסדרון אזי מימינו יהיה ערכי אפור נמוכים (כהה) ומשמאלו ערכי אפור גבוהים (בהיר) ובכך תוצאת הפילטר עבור פיקסל זה תהיה ערך חיובי גדול.

לאחר הרצת הפילטר אנו ממירים את התמונה המתקבלת לתמונת שחור לבן ע"י קביעת threshold לתמונה.

דוגמא להרצת הפילטר :



כפי שניתן לראות הפילטר גם מזהה מעברים של תמונה על קיר וכו' לכן נדרש סינון לפי מבנה מורפולוגי לכן אנו מחשבים לתמונת שחור לבן שהתקבלה את רכיבי הקשירות ובוחרים ברכיב בעל המאפיינים הבאים :

- מספר פיקסלים מעל 800.
- גובה מעל 200 פיקסל.
- מקיים תנאים הנ"ל והכי קרוב לקצה התמונה (הימיני בחלק הימיני ושמאל בחלק שמאל).

כתוצאה מניתוח זה של התמונה אנו מחזירים את העמודה של הדלת משמאל הקרובה ביותר לרובוט והעמודה של הדלת הימינית הכי קרובה לרובוט (אם אין מוחזר ערך המסמן שלא זוהתה דלת).

ספירת דלתות

על מנת להעריך את מיקומו של הרובוט במהלך הנסיעה במסדרון, נעשה שימוש בדלתות משני צידיו.
בעת זיהוי דלת כלשהי בתמונה, יש לקבוע האם זו דלת חדשה או דלת שזוהתה בעבר.
בנוסף, בעת זיהוי דלת חדשה, יש לבצע בדיקה האם הרובוט הגיע לדלת היעד.
במהלך נסיעתו של הרובוט לכיוון דלת מסוימת, קצה הדלת ילך ויתקרב מכיוון מרכז התמונה לכיוון קצות התמונה.
עובדה זו, מסייעת להכריע האם זיהינו את הדלת בעבר או שנתקלנו בדלת חדשה.
לצורך ביצוע האלגוריתם, יש לשמור עבור צד ימין וצד שמאל, את הנתונים הבאים:

- מיקום הדלת האחרונה שזוהתה (בציר x).
- מספר התמונות בהן זיהינו את הדלת הנוכחית.

<u>האלגוריתם:</u>

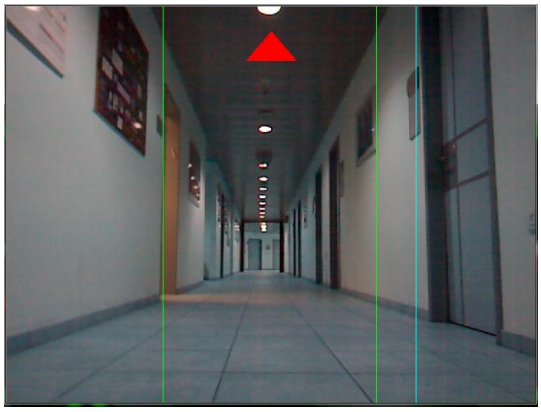
- כאשר מזהים דלת מצד שמאל:
  - בדוק האם מיקום הדלת הנוכחית מתקרב לצד השמאלי של התמונה ביחס לדלת הקודמת שזוהתה.
    - אם כן:
      - הגדל את מספר התמונות בהן זיהינו את הדלת הנוכחית ב-1.
      - עדכן את מיקום הדלת.
      - במידה ומספר התמונות בהן הדלת זוהתה גדול מסף מסוים והדלת קרובה מספיק לקצה השמאלי של התמונה, עדכן את מיקום הרובוט בהתאם למיקום הדלת.
      - אחרת, המשך בביצוע האלגוריתם.
    - אחרת:
      - זוהי דלת חדשה ולכן יש לאתחל את מיקומה, ולאתחל ל-1 את מספר התמונות בהן זיהינו את הדלת הנוכחית.
- כאשר מזהים דלת מצד ימין:
  - בדוק האם מיקום הדלת הנוכחית מתקרב לצד הימני של התמונה ביחס לדלת הקודמת שזוהתה.
    - אם כן:
      - הגדל את מספר התמונות בהן זיהינו את הדלת הנוכחית ב-1.
      - עדכן את מיקום הדלת.
      - במידה ומספר התמונות בהן הדלת זוהתה גדול מסף מסוים והדלת קרובה מספיק לקצה הימני של התמונה, עדכן את מיקום הרובוט בהתאם למיקום הדלת.
      - אחרת, המשך בביצוע האלגוריתם.
    - אחרת:
      - זוהי דלת חדשה ולכן יש לאתחל את מיקומה, ולאתחל ל-1 את מספר התמונות בהן זיהינו את הדלת הנוכחית.

הדגמת אופן פעולת האלגוריתם:

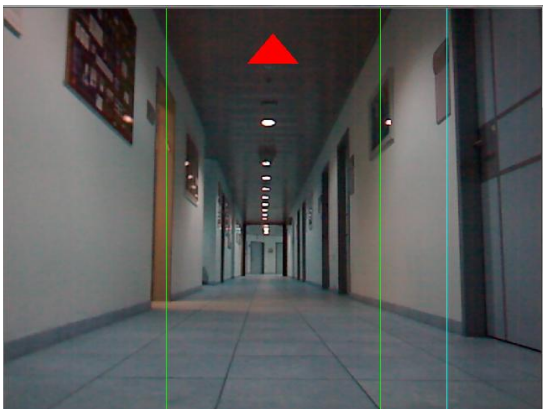
סימונים:

- מיקום הדלת מסומן בכחול.
- קצות המסדרון מסומנים בירוק.

זיהוי דלת בצד ימין של התמונה:

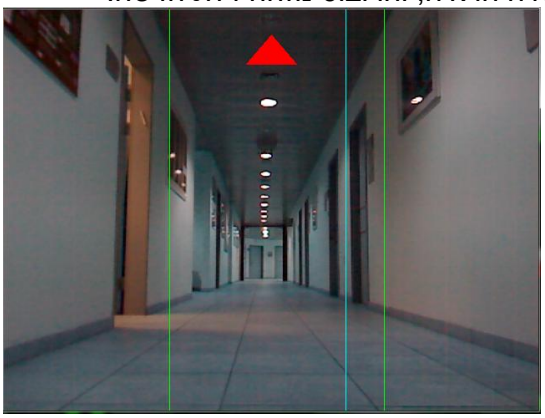


הדלת מתקרבת לקצה הימני של התמונה:



הדלת קרובה מספיק לקצה הימני של התמונה, וזוהתה במשך מספר תמונות רצופות ולכן מיקום הרובוט במפה מתעדכן לפיה:

הדלת שזוהתה יוצאת משדה הראיה, והרובוט מזהה דלת חדשה:

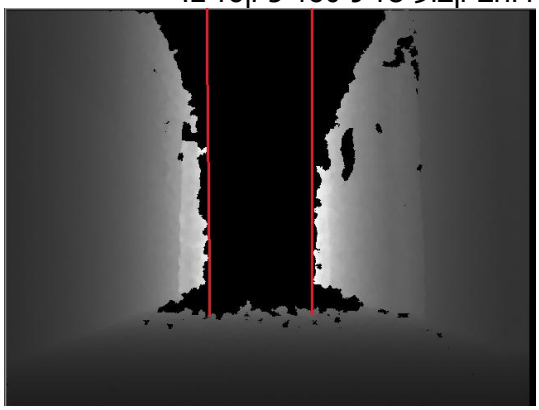


במהלך הנסיעה במסדרון, עבור כל צד, נשמר מספר הדלת הבאה שהרובוט צריך לזהות.
לאחר שהדלת זוהתה:

- עבור דלת שמאלית: מיקום הרובוט במפה מתעדכן לדלת השמאלית הבאה בתור.
- עבור דלת ימנית: מיקום הרובוט במפה מתעדכן לדלת הימנית הבאה בתור.

זיהוי יציאה מהמסדרון

במהלך הנסיעה במסדרון, יש לזהות את הרגע בו צריך לשנות את מצב הנסיעה לנסיעה בחלל.
לצורך כך נעשה שימוש בתמונות העומק, שמאפשרות חישוב של קצות המסדרון.
במהלך הנסיעה, למסדרון רוחב קבוע של כ-150 פיקסלים:

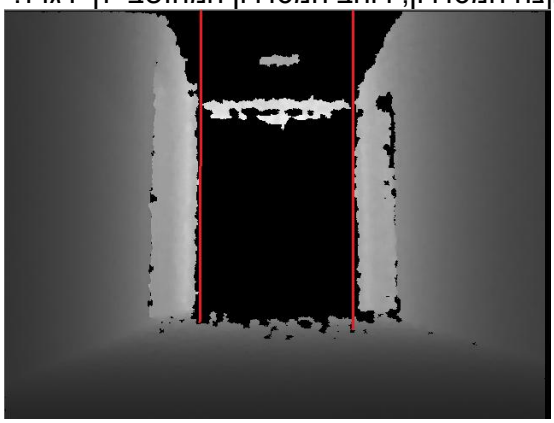


רוחב המסדרון מחושב ע"י סכימת כמות הפיקסלים השחורים בכל עמודה, וזיהוי הקצה השמאלי והימני של המסדרון כפי שמודגש באדום בתמונה.

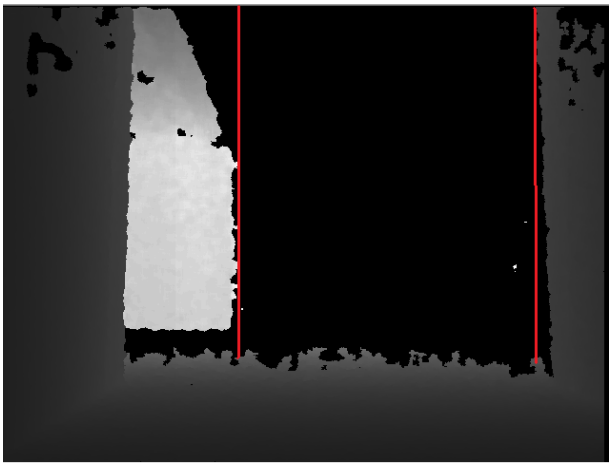
<u>האלגוריתם:</u>

- עבור כל עמודה בתמונה:
  - סכום את כמות הפיקסלים השחורים.
- עבור על העמודות מצד שמאל לצד ימין:
  - אם מספר הפיקסלים השחורים בעמודה גדול מקבוע גובה המסדרון:
    - במידה והקצה השמאלי של המסדרון עדיין לא נמצא, אתחל את הקצה השמאלי של המסדרון לעמודה זו.
    - אחרת, הגדל ב-1 את רוחב המסדרון.
  - אחרת:
    - במידה ורוחב המסדרון גדול מקבוע הרוחב המינימאלי (כ-60 פיקסלים), אתחל את הקצה הימני של המסדרון לעמודה זו וסיים.
    - אחרת, התחל מחדש לחפש קצה שמאלי.

כאשר הרובוט מתקרב לקצה המסדרון, רוחב המסדרון המחושב ילך ויגדל:



בשלב מסוים, רוחב המסדרון יהיה גדול מקבוע המקסימאלי המותר לנסיעה במסדרון (כ-320 פיקסלים):

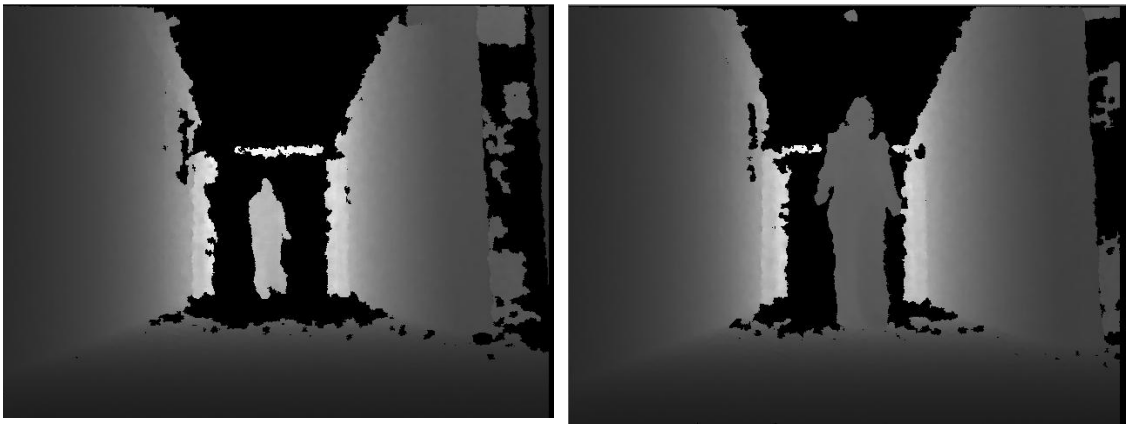


במצב זה, הרובוט יצא מהמסדרון ומצב הנסיעה ישתנה לנסיעה בחלל.

זיהוי מכשולים

במהלך הנסיעה במסדרון, הרובוט צריך להתמודד עם מצב בו מכשול מפריע להתקדמותו.
כאשר הוא מזהה מכשול בדרכו, עליו להמתין עד שהמכשול יזוז.
על מנת לזהות מצבים כאלה, נעשה שימוש בתמונת העומק ובאלגוריתם שתואר למציאת המסדרון.
במצב רגיל, בו אין מכשולים בדרך, המסדרון נראה באופן הבא:

כאשר יהיה מכשול בדרך, האלגוריתם שתואר ייכשל במציאת המסדרון:

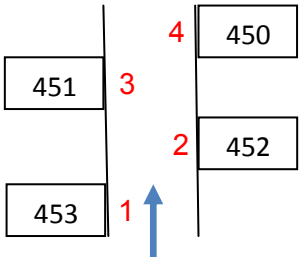


בדוגמאות אלו, לא יימצא רצף של פיקסלים שחורים ברוחב מספיק על מנת שיזוהה כמסדרון.

אלגוריתם תיקון מיקום הרובוט במצב בו דלת לא מזוהה

במהלך הנסיעה במסדרון, קיימים מצבים בהם חלק מהדלתות לא מזוהות לאורך מספר גדול מספיק של תמונות (למשל בגלל תאורה לא מספיקה או פניה חדה של הרובוט).
בכדי להתמודד עם מצבים אלו ולעדכן את מיקום הרובוט בהתאם, הוכנס אלגוריתם לתיקון פספוס דלתות.
עבור כל דלת, נשמר המספר הסידורי בו היא צריכה להופיע, על-פי סדר הזיהוי של הרובוט בנסיעה (המספר הסידורי לא תמיד חופף למספור הדלתות בפועל).

לדוגמא:



המספר הסידורי מסומן באדום. כלומר הרובוט צפוי לזהות תחילה את דלת 453, לאחר מכן את דלת 452, לאחר מכן 451 ובסוף 450.

במהלך הנסיעה במסדרון, עבור כל צד נשמר המספר הסידורי של הדלת הבאה אותה הרובוט צריך לזהות.
בעת זיהוי דלת בצד שמאל, מתבצע במידת הצורך עדכון של המספר הסידורי של הדלת הבאה מצד ימין (במידה ומספר זה קטן מהמספר הסידורי של הדלת שזוהתה כעת בצד שמאל).
באופן דומה, מבצעים את העדכון בעת זיהוי דלת בצד ימין.
מצב זה מונע עדכון שגוי של מיקום הרובוט במידה ודלת מסוימת לא מזוהה.

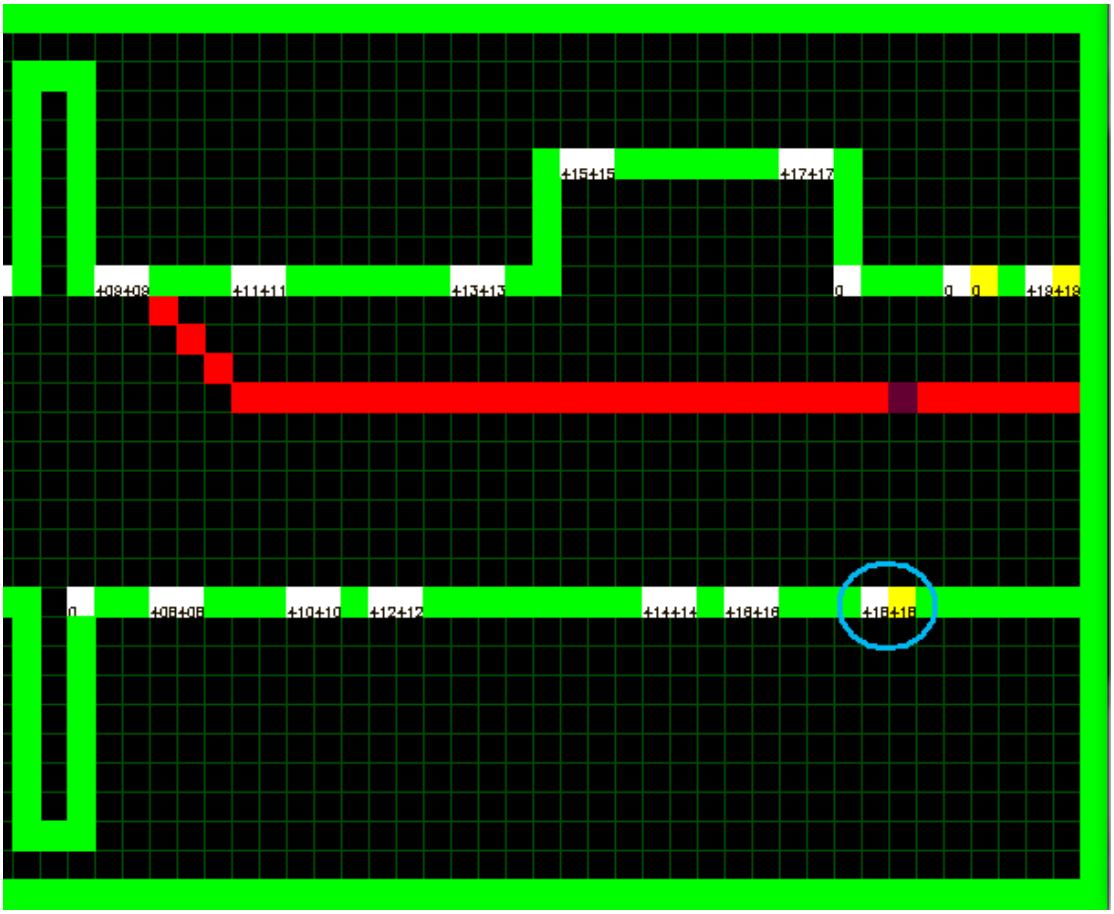
לדוגמא:
נניח שהרובוט נכנס למסדרון שמתואר באיור.

- מספר הדלת הסידורי הבא מצד שמאל הוא 1.
- מספר הדלת הסידורי הבא מצד ימין הוא 2.
- הרובוט מזהה את דלת מספר 453 ולכן מספר הדלת הסידורי הבא מצד שמאל מתעדכן ל-3.
- הרובוט לא מזהה את דלת מספר 452.
- הרובוט מזהה את דלת מספר 451. מכיוון שדלת מספר 452 לא זוהתה, המספר הסידורי של הדלת הבאה מצד ימין קטן מהמספר הסידורי בצד שמאל ולכן מספר זה מתעדכן להיות 4. כלומר הרובוט כעת יצפה לדלת 450 מצד ימין ולא לדלת 452 אותה פספס.
- הרובוט מזהה את דלת מספר 450 והמיקום במפה מתעדכן בצורה טובה למרות שהדלת הקודמת לא זוהתה.

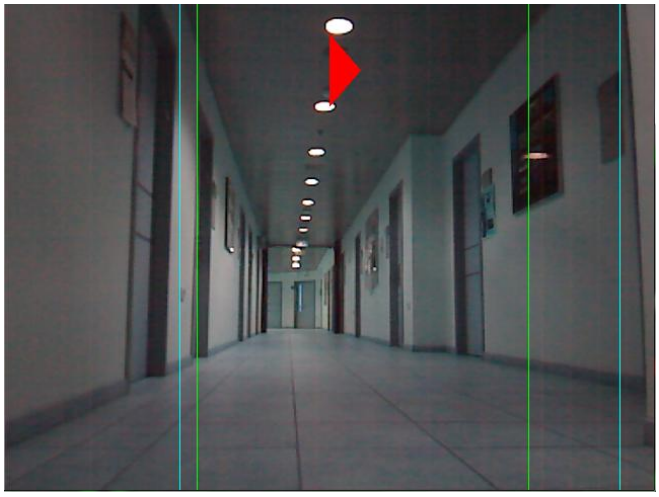
<u>הדגמת אופן פעולת האלגוריתם:</u>

דלת מספר 418 מזוהה:

המיקום במפה מתעדכן:

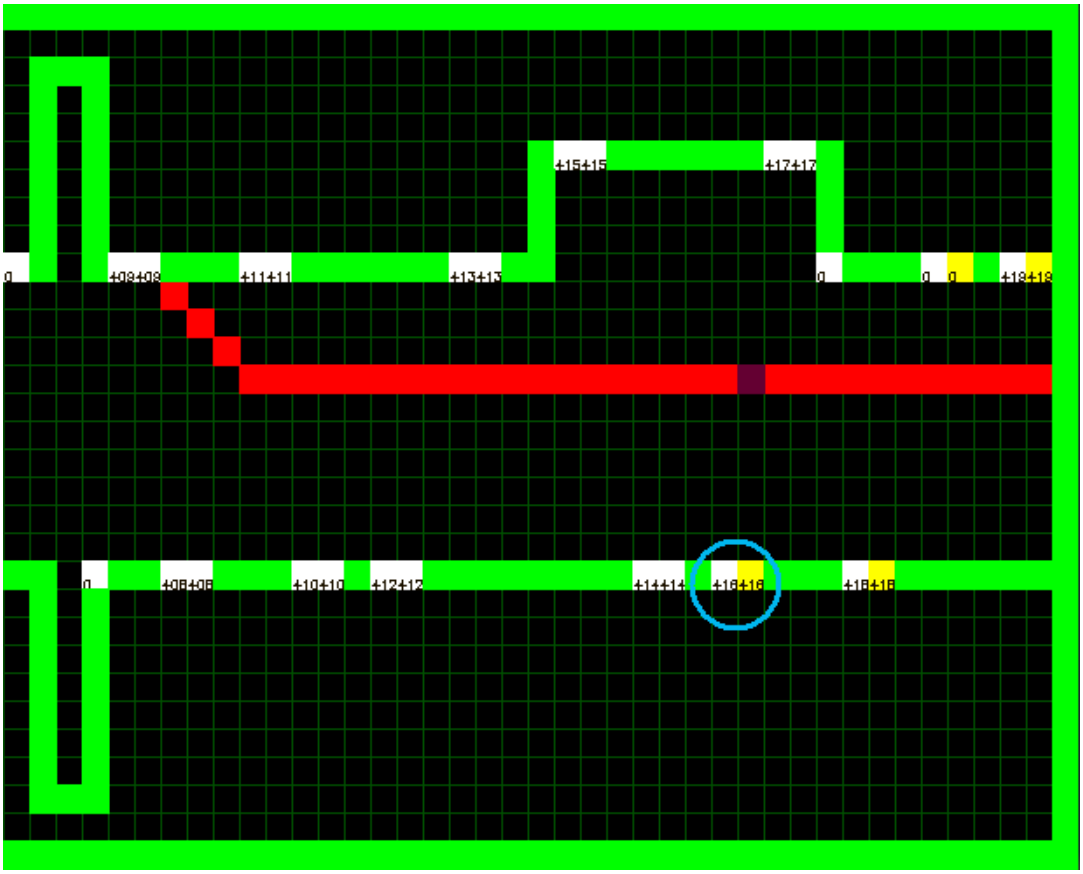


עקב פניה חדה של הרובוט על מנת להתיישר למרכז המסדרון, דלת מספר 417 לא מזוהה למשך מספר גדול מספיק של תמונות עוקבות.



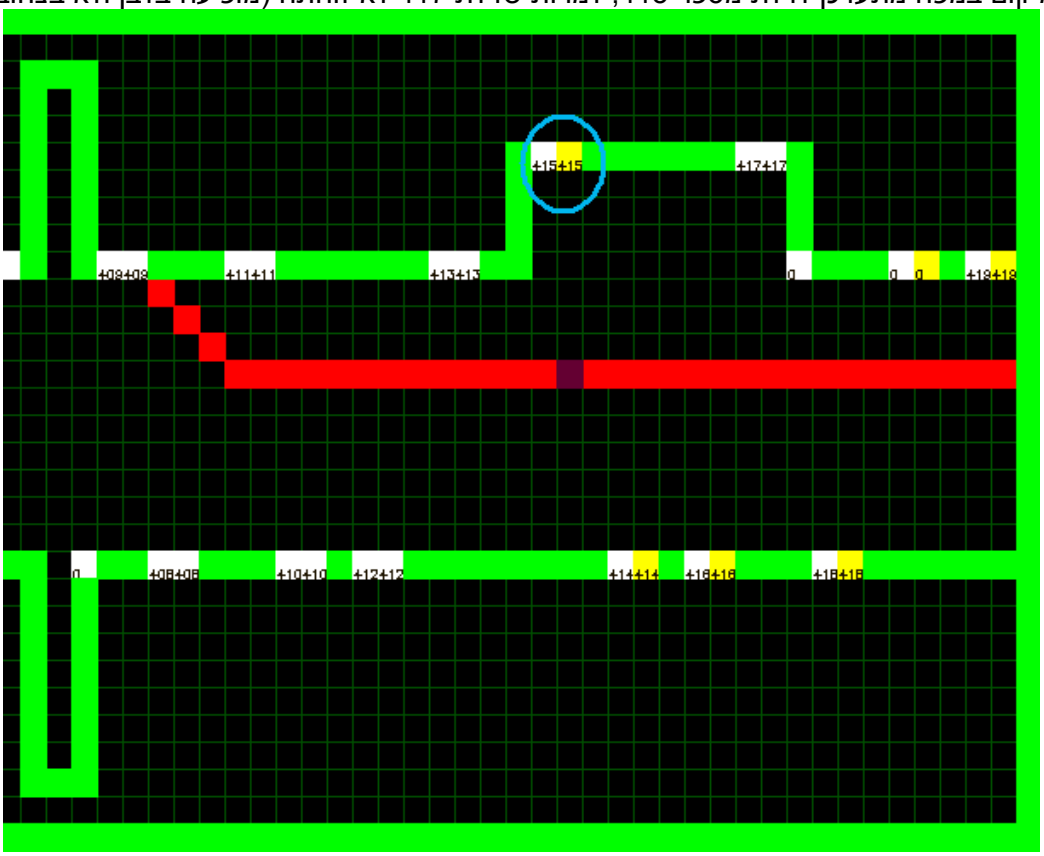
דלת מספר 416 מזוהה:

המיקום במפה מתעדכן, ואלגוריתם התיקון מעדכן את מספר הדלת שצפויה להתגלות מצד ימין ל-415 למרות שבפועל דלת 417 כלל לא זוהתה.



הרובוט מזהה את דלת 414, ולאחר מכן את דלת 415:

המיקום במפה מתעדכן לדלת מספר 415, למרות שדלת 417 לא זוהתה (מופיעה בלבן ולא בצהוב):



<u>הערה:</u> האלגוריתם לא מתמודד עם מצב בו דלת מצד ימין ודלת מצד שמאל לא זוהו ברצף (לא יהיה ניתן לבצע את התיקון על פי צד כלשהו).

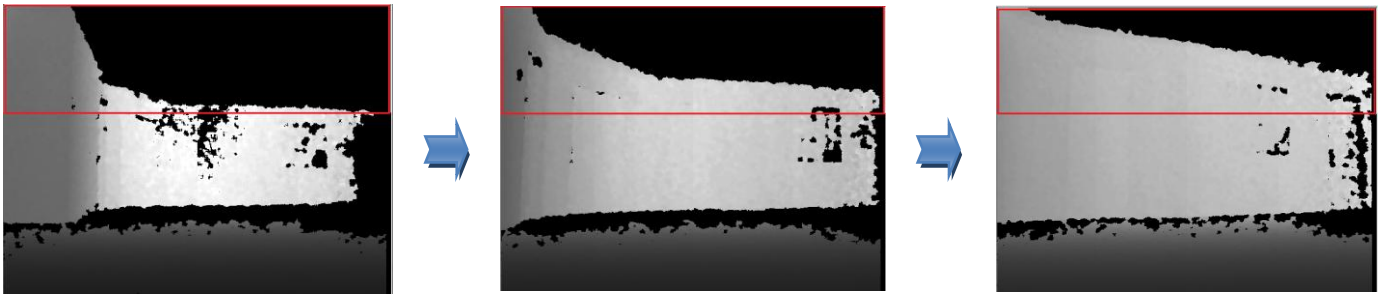
נסיעה בחלל

חישוב זוויות הפניה

על מנת להגיע ממסדרון אחד למסדרון השני, הרובוט צריך לעבור בחלל פתוח, ולבצע פניה ימינה או פניה שמאלה.

פניה שמאלה	פניה ימינה
	

על מנת לחשב את זווית הפניה לאורך הנסיעה בחלל, כך שהרובוט יגיע מקצה מסדרון אחד לתחילת המסדרון השני נעשה שימוש בתמונות העומק.
לצורך כך, נעשה שימוש בתופעה שככל שמתקדמים לעבר הקיר שנמצא מול היציאה מהמסדרון, מספר הפיקסלים השחורים בחלק העליון של התמונה קטן.



לאחר סכימת מספר הפיקסלים השחורים בתמונה, מתבצע חישוב פקטור הפניה באופן הבא:

$$factor = \frac{1-\left(\left(black\_pixels - MIN\_BLACK\right)/\left(MAX\_BLACK - MIN\_BLACK\right)\right)}{turn\_factor}$$

כאשר:

- MAX_BLACK - קבוע שמתאר את מספר הפיקסלים השחורים המקסימאלי שניתן לראות במהלך הפניה (חושב על-פי מדגם מייצג של תמונות).
- MIN_BLACK - קבוע שמתאר את מספר הפיקסלים השחורים המינימאלי שניתן לראות במהלך הפניה (חושב על-פי מדגם מייצג של תמונות).
- $turn\_factor$ - קבוע שמחושב על-פי עוצמת הפניה שצריכה להתבצע, כאשר קיים הבדל בין פניה ימינה לפניה שמאלה.

לאחר חישוב פקטור הפניה, מתבצע תרגום לפקודות תנועה עבור הרובוט.
עבור פנייה ימינה:

- מהירות הגלגל השמאלי: $factor * 256$.
- מהירות הגלגל הימני: 256.

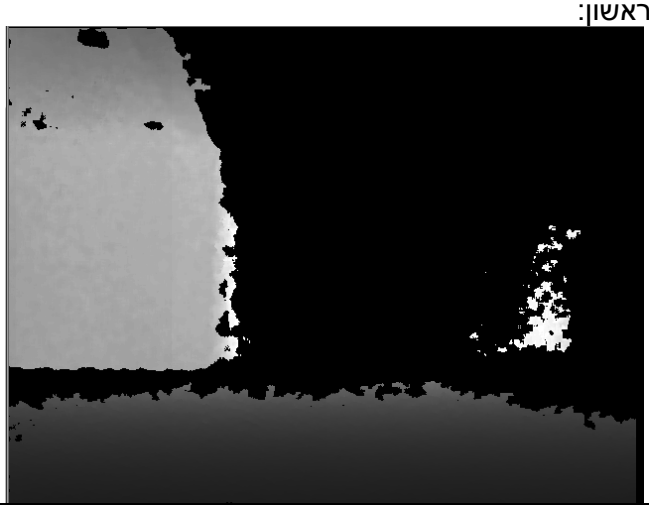
עבור פנייה שמאלה:

- מהירות הגלגל השמאלי: 256.
- מהירות הגלגל הימני: $factor * 256$.

ביציאה מהמסדרון הראשון הפקטור המחושב יהיה קטן מכיוון שיש מספר גדול של פיקסלים שחורים בחלק העליון של התמונה. לכן, עפ"י הנוסחא, עוצמת הפניה תהיה קטנה.
ככל שמתקרבים לקיר שנמצא מול המסדרון, מספר הפיקסלים השחורים קטן והפקטור המחושב גדל. לכן, עוצמת הפניה תגדל והרובוט יפנה לכיוון מרכז המסדרון השני.

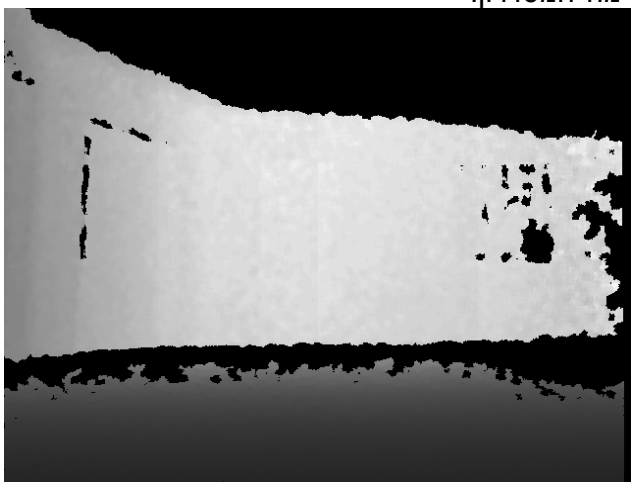
נדגים זאת בעזרת מספר תמונות שנדגמו במהלך פניה ימינה:

ביציאה מהמסדרון הראשון:



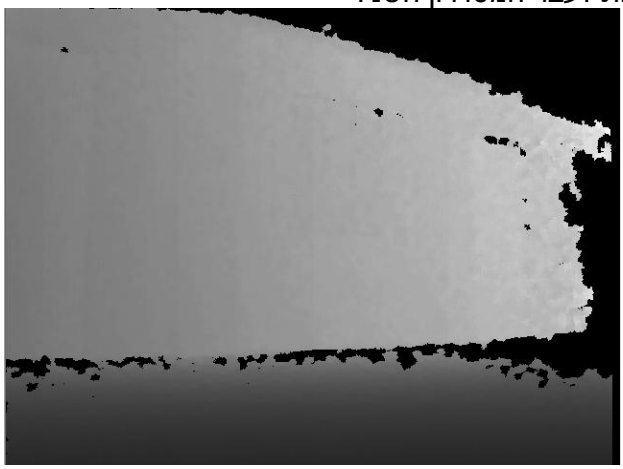
מספר הפיקסלים השחורים הוא 70225.
מהירות הגלגל השמאלי: 289.
מהירות הגלגל הימני: 256.

כאשר מתקרבים לקיר מול המסדרון:



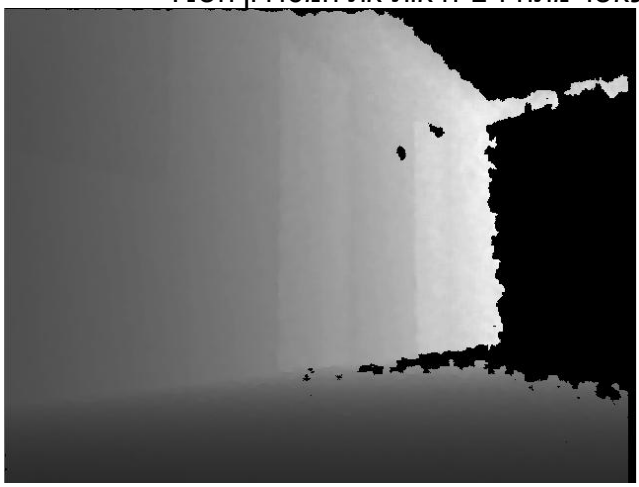
מספר הפיקסלים השחורים הוא 61289.
מהירות הגלגל השמאלי: 300.
מהירות הגלגל הימני: 256.

לאחר שמתחילים לפנות לעבר המסדרון השני:



מספר הפיקסלים השחורים הוא 28293.
מהירות הגלגל השמאלי: 337.
מהירות הגלגל הימני: 256.

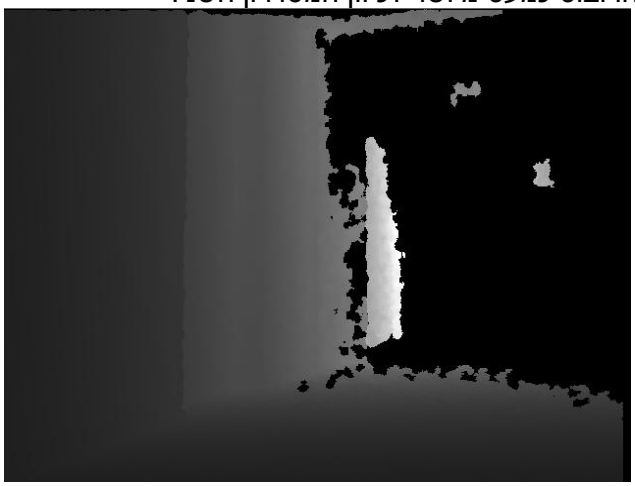
לקראת סיום הפניה, כאשר מתחילים לראות את המסדרון השני:



מספר הפיקסלים השחורים הוא 33086.

מהירות הגלגל השמאלי: 332.
מהירות הגלגל הימני: 256.

בסיום הפניה, כאשר הרובוט כמעט מיושר לכיוון המסדרון השני:



מספר הפיקסלים השחורים הוא 52777.
מהירות הגלגל השמאלי: 309.
מהירות הגלגל הימני: 256.

השימוש באלגוריתם שתואר לצורך חישוב זווית הפנייה, יגרום לרובוט להגיע אל מול המסדרון השני.

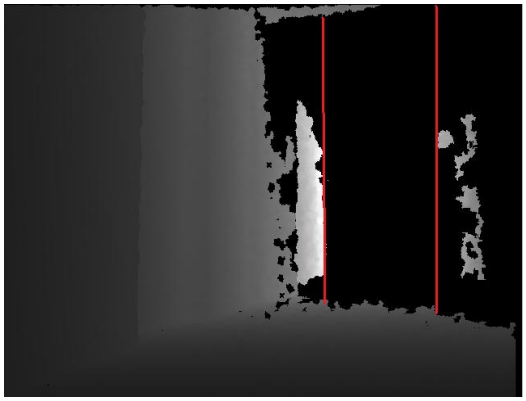
זיהוי כניסה למסדרון

במהלך הנסיעה בחלל, יש לזהות את הרגע בו צריך לשנות את מצב הנסיעה לנסיעה במסדרון (כאשר הרובוט מסיים את הפנייה ומתייצב אל מול המסדרון).
לצורך כך נעשה שימוש בתמונות העומק, שמאפשרות חישוב של קצות המסדרון.
במהלך הנסיעה בחלל, עבור כל תמונה, מחושב הקצה הימני והקצה השמאלי של המסדרון על-פי האלגוריתם שתואר במצב נסיעה במסדרון.

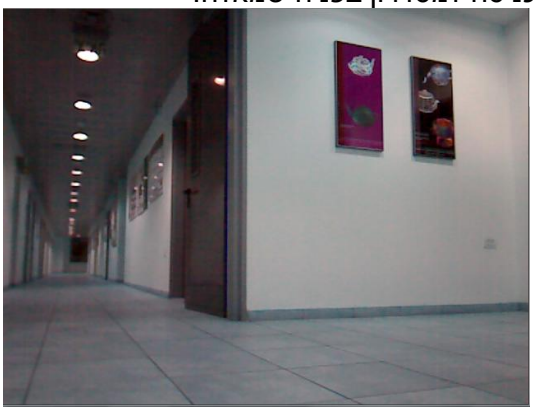
- עבור פנייה ימינה:
  הרובוט יעבור למוד נסיעה במסדרון כאשר זוהה מסדרון ברוחב מתאים שהקצה הימני שלו ממוקם בחלק הימני של התמונה (בין עמודה 540 ל-620).
- עבור פנייה שמאלה:
  הרובוט יעבור למוד נסיעה במסדרון כאשר זוהה מסדרון ברוחב מתאים שהקצה השמאלי שלו ממוקם בחלק השמאלי של התמונה (בין עמודה 20 ל-100).

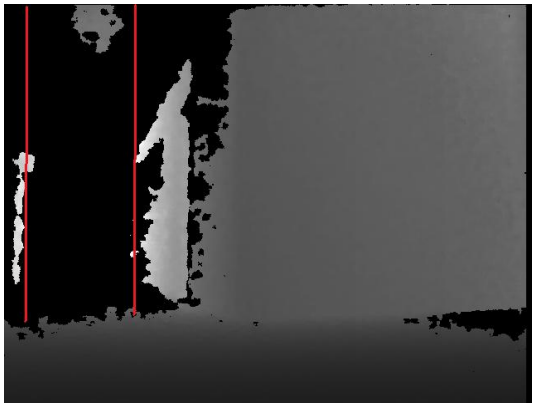
לאחר שהכניסה למסדרון זוהתה, הרובוט מסיים לפנות לעבר המסדרון ועובר למצב נסיעה במסדרון שתואר בפרק הקודם.

כניסה למסדרון בפניה ימינה:



כניסה למסדרון בפניה שמאלה:





תצוגת מפה

פורמט קובץ אתחול

על מנת לאתחל לרובוט את המפה שבעזרתה ייבצע את הניווט נעשה שימוש בקובץ בעל המבנה הבא:

- השורה הראשונה מציינת את אורך ורוחב המפה.
- לאחר מכן כל שורה מציינת אלמנט במפה: קיר או דלת.
- בין החלקים השונים במפה (מסדרון\חלל) ישנה שורת הפרדה.

<u>דוגמא לשורה ראשונה בקובץ:</u>

$$(80,30)$$

כלומר רוחב המפה הוא 80 תאים וגובה המפה הוא 30 תאים.

<u>דוגמא לשורת קיר:</u>

$$(10,0,1)$$

כאשר 10 מציין את קואורדינאטת ה-X של הקיר, 0 מציין את קואורדינאטת ה-Y, ו-1 מציין שזוהי הגדרה של קיר.

<u>דוגמא לשורת דלת:</u>

$$(9,11,2,454,13,1)$$

כאשר 9 מציין את קואורדינאטת ה-X של הדלת, 11 מציין את קואורדינאטת ה-Y, 2 מציין שזוהי הגדרה של דלת, 454 מציין את מספר החדר שהדלת מייצגת, 13 מציין את המספר הסידורי של הדלת (לפי סדר הזיהוי שהרובוט צפוי לזהות את הדלתות מכיוון החלל לכיוון עומק המסדרון) ו-1 מציין שהדלת ממוקמת בצד ימין מכיוון החלל לכיוון עומק המסדרון (0 יציין דלת בצד שמאל) .

<u>דוגמא לשורת הפרדה:</u>

$$(0,0,0)$$

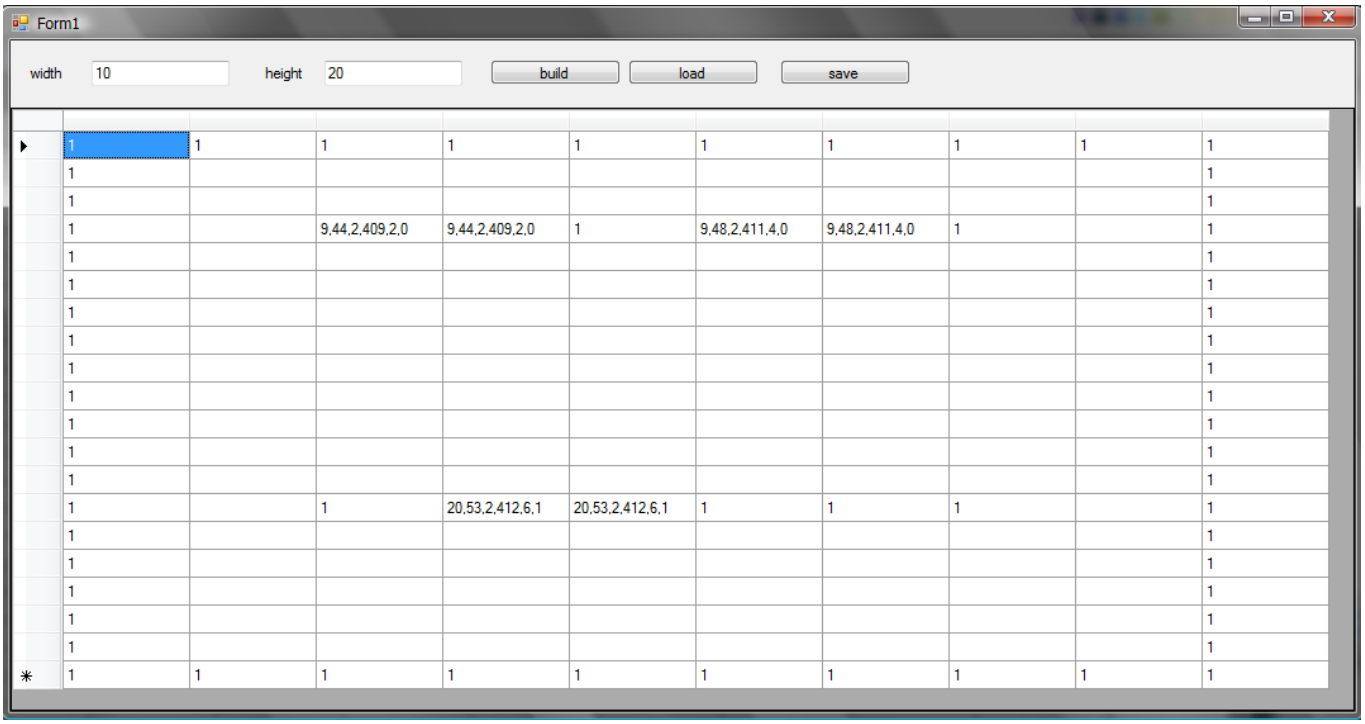
קובץ האתחול יכיל 3 חלקים, כאשר בין חלק לחלק תהיה שורת הפרדה:

- מסדרון ראשון: יכיל הגדרות של קירות ודלתות.
- חלל: יכיל הגדרות של קירות.
- מסדרון שני: יכיל הגדרות של קירות ודלתות.

תוכנה ליצירת קובץ האתחול

על מנת ליצור את קובץ האתחול בצורה נוחה יותר, נבנה ממשק וויזואלי שמאפשר לקבוע את גודל המפה הרצויה ולהזין את האובייקטים לתאים במטריצה שנוצרה (קיר\דלת).
לאחר מכן, הקובץ ייווצר לפי ההגדרות באופן אוטומטי.

דוגמא:



▸	1	1	1	1	1	1	1	1	1	1
	1									1
	1									1
	1		9,44,2,409,2,0	9,44,2,409,2,0	1	9,48,2,411,4,0	9,48,2,411,4,0	1		1
	1									1
	1									1
	1									1
	1									1
	1									1
	1									1
	1									1
	1									1
	1									1
	1		1	20,53,2,412,6,1	20,53,2,412,6,1	1	1	1		1
	1									1
	1									1
	1									1
	1									1
	1									1
*	1	1	1	1	1	1	1	1	1	1

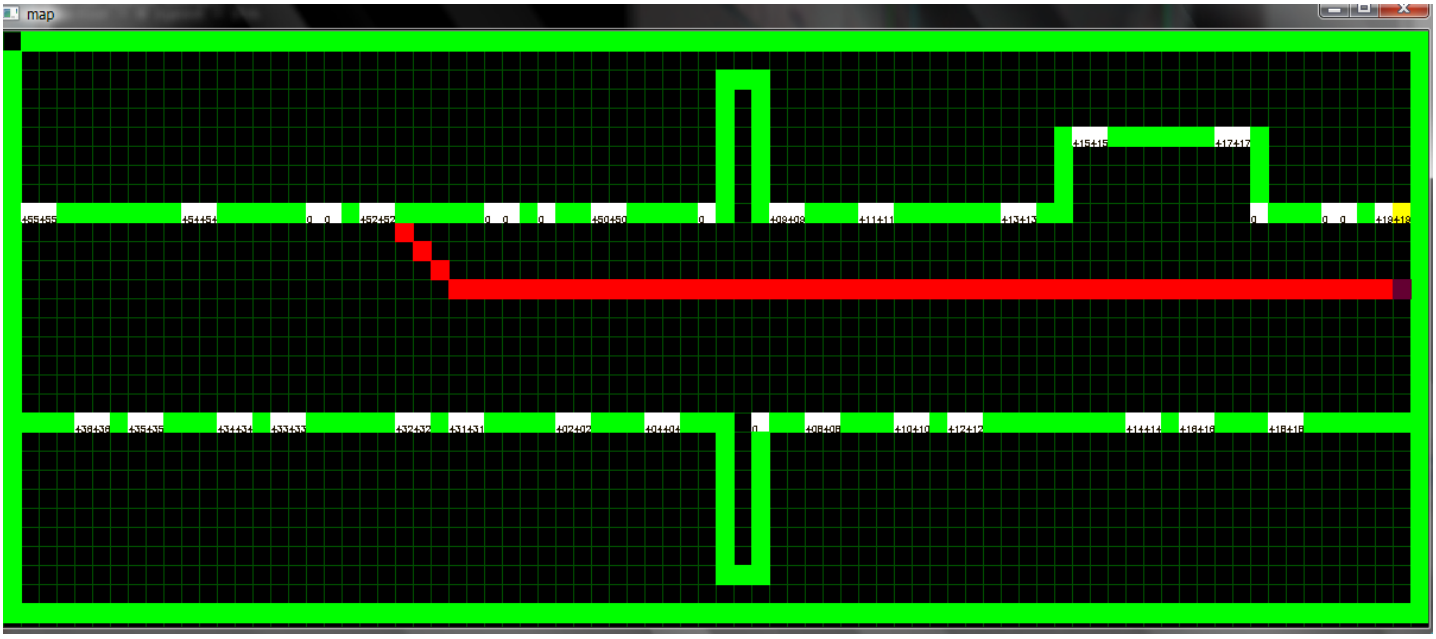
בדוגמא זו מוגדרות במפה 3 דלתות: 409, 411,412 כאשר כל דלת ברוחב 2 תאים.

מבנה התצוגה

במהלך ניווט הרובוט מחדר המקור לחדר היעד, ניתן לצפות בתצוגת המפה על מנת לדעת את מיקומו המשוערך.
התצוגה מחולקת לשני חלקים:

- מפה: מתארת את מבנה הקומה, מסלול הניווט, חדר המוצא וחדר היעד ומיקום הרובוט הנוכחי.
- תמונה מעובדת נוכחית: מתארת את התמונה האחרונה שהרובוט עיבד, הכוללת מידע על גבולות המסדרון וזיהוי הדלתות.

<u>מפה</u>

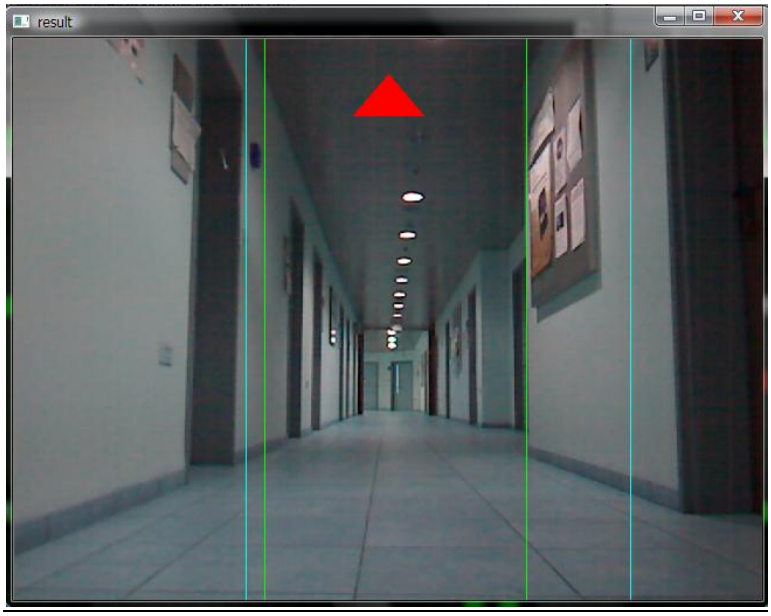


מפה זו מתארת את מבנה קומה 4 בבניין טאוב.
בקומה זו שני מסדרונות, במרכזם חלל פתוח.

סימונים:

- מסלול הרובוט: מתואר בצבע אדום. בדוגמא עליו להגיע מחדר 419 לחדר 452.
- מיקום הרובוט הנוכחי: מתואר בצבע סגול. בדוגמא דלת 419.
- קירות: מסומנים בצבע ירוק.
- דלתות שטרם זוהו: מסומנות בצבע לבן, ומספר החדר מצוין עליהן.
- דלתות שזוהו: מסומנות בצבע צהוב.

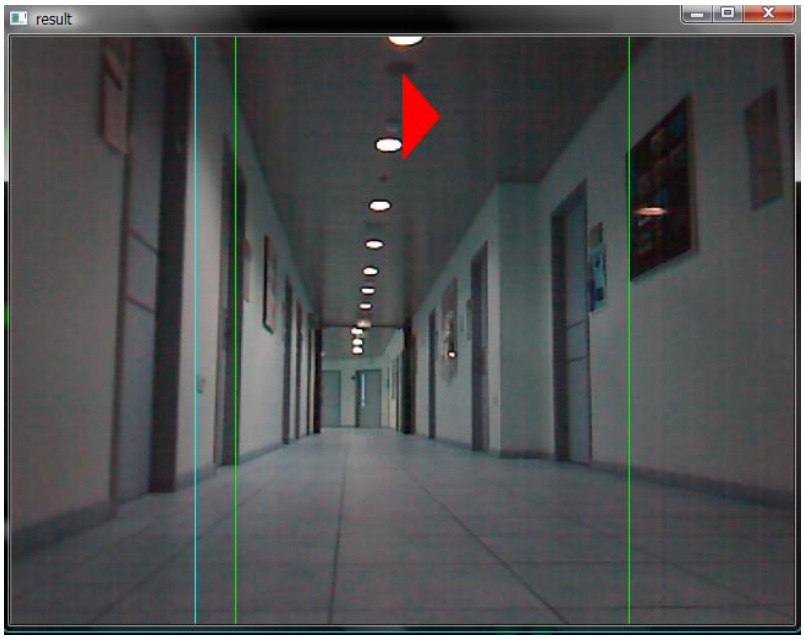
תמונה מעובדת



סימונים:

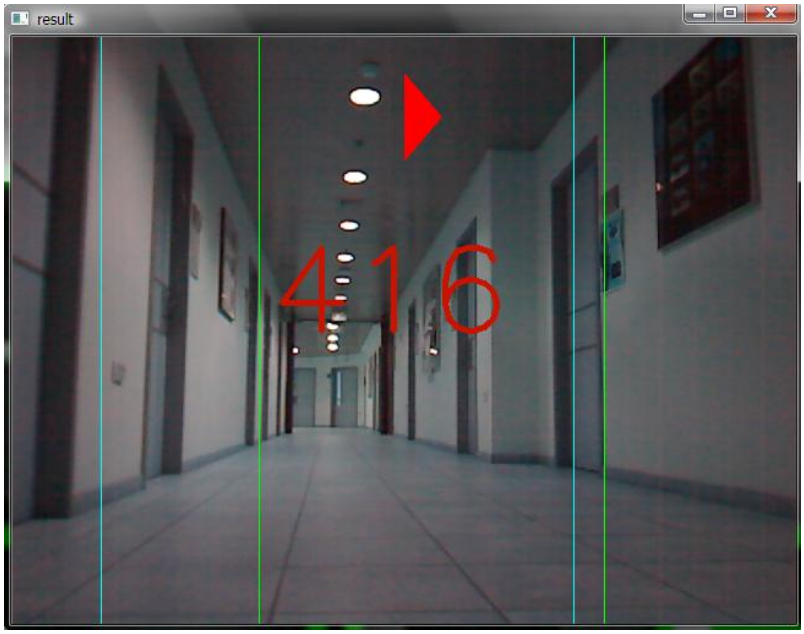
- חץ אדום: מתאר את כיוון הפניה של הרובוט. בדוגמא הרובוט נוסע ישר מאחר ונמצא במרכז המסדרון.
- קוים אנכיים ירוקים: מתארים את גבולות המסדרון.
- קוים אנכיים כחולים: מתארים את זיהוי הדלתות מצד ימין וצד שמאל.

דוגמא לתצוגה במהלך פנייה של הרובוט ימינה:

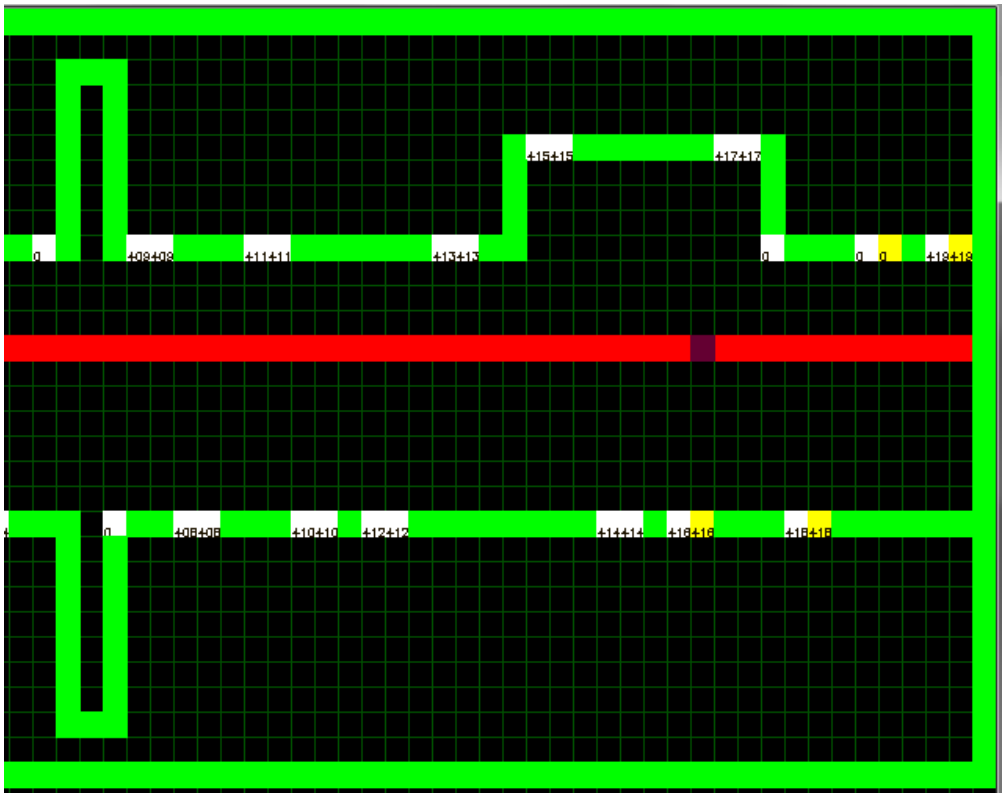


ניתן לראות שהחץ האדום מצביע כעת ימינה.

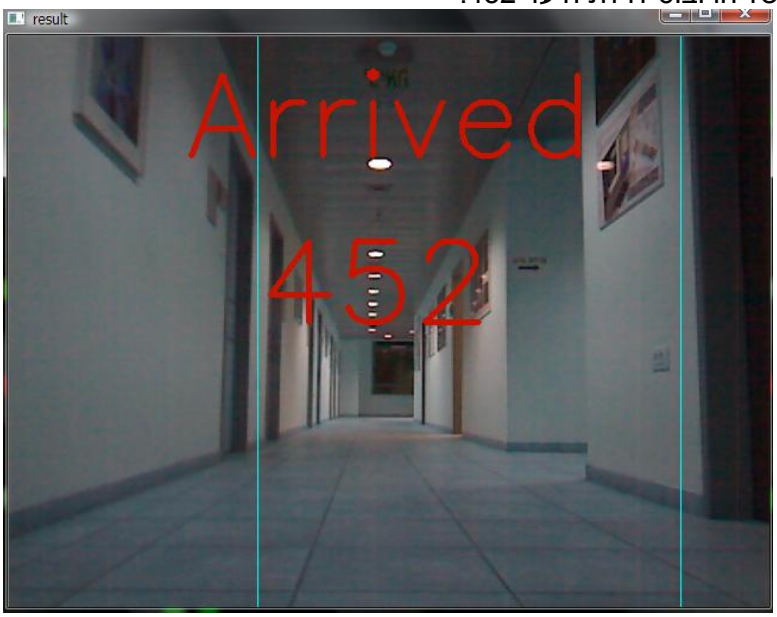
דוגמא לתצוגה בעת זיהוי דלת מספר 416 מצד שמאל:



במפה דלת זו נצבעה לצהוב ומיקום הרובוט על המסלול התעדכן:



דוגמא להגעה של הרובוט לדלת היעד 452:



תוצאות

על מנת למדוד את נכונות תוצאות האלגוריתמים השתמשנו בשיטת המדידה precision-recall, כאשר

Precision – יחס של העצמים שהובאו שהם רלוונטים.

Recall – יחס של העצמים הרלוונטים מתוך העצמים הובאו.

הבדיקה נערכה על 2 הרצות המכילות כ 1000 תמונות.

נחלק את המערכת לשני תת מערכות :

<u>אלגוריתמים לקביעת עצם שהוא דלת (מערכת הראייה)</u>

נבדוק בחלק זה את הprecision-recall כאשר בתמונה כלשהי עצם רלוונטי הוא **דלת כלשהי**.

Recall – אחוז הדלתות מתוך העצמים שהאלגוריתם החזיר.

Precision – אחוז העצמים שהאלגוריתם טען שהם דלת ואכן הם דלת.

Recall = 100%

Precision=86%

<u>אלגוריתמים לקביעת דלת כחדשה</u>

נבדוק בחלק זה את הprecision-recall כאשר עצם רלוונטי הוא דלת **במפה**.

Recall – אחוז הדלתות במפה מתוך העצמים שהאלגוריתם החזיר.

Precision – אחוז העצמים שהאלגוריתם טען שהם דלת חדשה ואכן הם דלת חדשה.

Recall = 100%

Precision = 90%

מסקנות :

- מערכת הראיה מביאה את כל הדלתות לזיהוי אך קיימים עצמים שהמערכת טוענת שהם דלתות אך הם לא (למשל מסדרונות צידיים).
- על מנת להתמודד עם חוסר הדיוק של אלגוריתם קביעת הדלתות כחדשות הוכנסו המסדרונות הצידיים כדלתות למפה.

נספחים

מיקום קבצי הקוד

1. תחת תיקיית C:\Shared\Navigation
   a. Navigation – ספריה המכילה את הקוד שמשמש לאלגוריתם הניווט.
   b. MapDisplayer – ספריה המכילה את הקוד שמשמש לתצוגה.
   c. Maps – תיקייה המכילה את קבצי המפה.

הוראות הפעלה של התוכנית

1. בשימוש בשלט :
   a. הפעלת האפליקציה – לחץ
   b. אתחול חדר מקור – לחץ על 3 ספרות של החדר.
   c. אתחול חדר יעד – לחץ על 3 ספרות של החדר.
   d. הפעלת אלגוריתם הניווט – לחץ 2.
   e. השהיית אלגוריתם הניווט (עצירת הרובוט) – לחץ 1.
   f. סגירת האפליקציה – לחץ 4.
2. התחברות מרחוק
   a. התחברות דרך VNCViewer –
      i. Ip : 192.168.1.102
      ii. שם משתמש : atom
      iii. סיסמא : atomvnc