



# מערכת SLAM לרכב אוטונומי

פרויקט מספר: 24-1-1-3049

מבצעים:

תובל

אורית

מנחה: רועי רייך

מקום ביצוע הפרויקט: אוניברסיטת תל אביב

# נושא הפרויקט



תמונת הרכב האוטונומי באוניברסיטה

- Simultaneous Localization and Mapping – SLAM

- שימוש בחיישנים למיפוי סביבת הרכב

- מפה מתעדכנת של הסביבה

- מיקום הרכב בתוך המפה

- הטמעה והרצת אלגוריתמי SLAM בסביבת Ubuntu 20.04 תוך שימוש

- בתוכנת Ros2 Foxy במחשבים ביתיים ומחשבי הרכב

# מטרות הפרויקט והמוטיבציה ההנדסית

## ❖ מטרות הפרויקט:

- סקירה של אלגוריתמי SLAM שיתאימו לעבודה בסביבת הרכב האוטונומי
- הטמעת אלגוריתמים מתאימים במחשב ביתי
- הטמעת האלגוריתמים במחשבים והרצתם תוך שימוש בחיישני LiDAR ו-INS\IMU הנמצאים ברכב

## ❖ המוטיבציה:

- חקר אלגוריתמי SLAM בסביבת עבודה משולבת ROS
- לחקור את יעילות אלגוריתמי SLAM מבוססי ליידר במכוניות אוטונומיות

# שיטות ומימוש

- **הרכב מערכת הרכב:**

- חיישני LiDAR – טכנולוגיה אופטית בעזרתה האלגוריתם מקבל נתוני מרחק מהסביבה
- חיישני INS\IMU – חיישן משולב בעזרתו מקבלים מידע על מיקום ומהירות המערכת
- מערכת ROS –

\* תוכנה בעזרתה מייצרים סביבת פיתוח נוחה לרובוטים

\* ניתן לפתח כל חלק ברובוט בנפרד

\* בעלת כלי ויזואלי Rviz המאפשר לבצע סימולציות

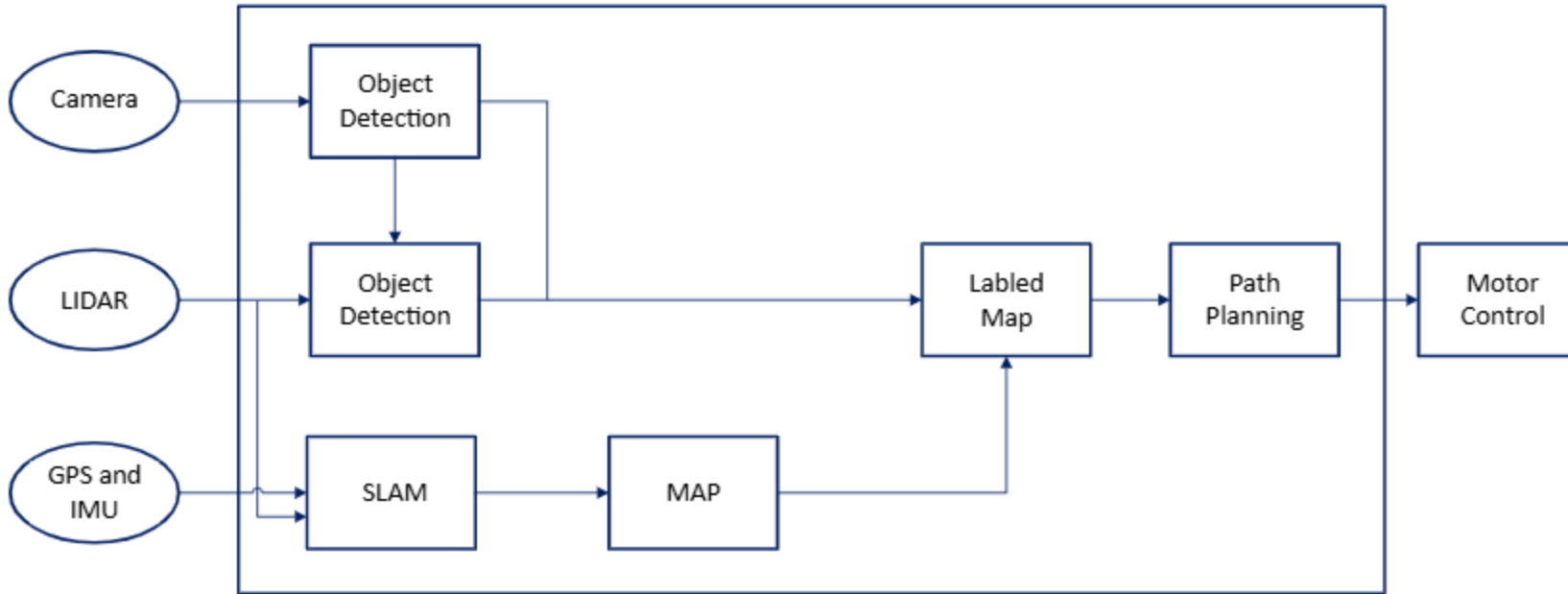
- **בדיקות האלגוריתמים:**

- הרצת האלגוריתמים במחשבים הביתיים על גבי מאגרי נתונים
- הרצת האלגוריתמים במחשבי הרכב על גבי נתוני החיישנים



חיישן OUSTER המובנה ברכב האוטונומי

# דיאגרמת בלוקים



## :SLAM

- משתמש בנתוני החיישנים
- ממפה את הסביבה
- ממקם את הרכב במפה

## :MAP

- תוצר ישיר של ה-SLAM
- נוצר מהודעות Point Cloud ו-Path בROS2

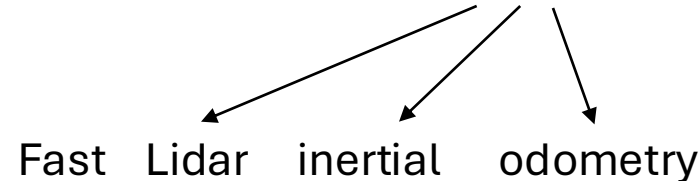
## :Labeled Map

- מפה מתעדכנת בזמן אמת
- תיוגים לעצמים במרחב

## :Path Planning

- קביעת נקודת הגעה במרחב
- ניווט אוטונומי לנקודה

# הסבר על Fast Lio

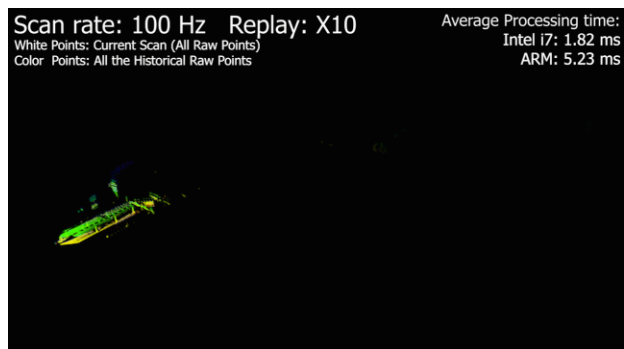


## עקרונות פעולה:

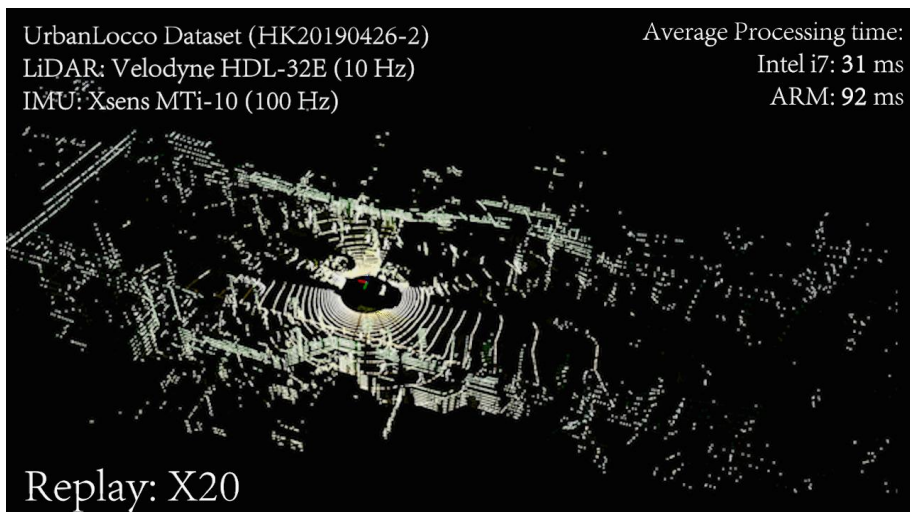
- האלגוריתם מקבל נתוני ליידר ונתונים אינרציאליים ובעזרתם משערך את מיקום הרכב
- אלגוריתם SLAM מבוסס מסנן

## יתרונות האלגוריתם:

- נתוני ה Point Cloud לא עוברים עיבוד – תורם למהירות האלגוריתם ולתיאור מדויק יותר של הסביבה
- הנתונים מתעדכנים באמצעות מבנה ikd tree – מאפשר זמני חישוב קצרים יותר
- תומך במגוון חיישני ליידר כגון : Ouster, Velodyne, Livox avia



דוגמאות הרצאה מהאינטרנט של Fast Lio



דוגמאות הרצאה מהאינטרנט של Fast Lio



# הדגמת סימולציות של Fast Lio



סרטון מתוך הרצת הסימולציה של Fast Lio במחשב הביתי

# הסבר על אלגוריתם LIORF

## מהו LIORF?

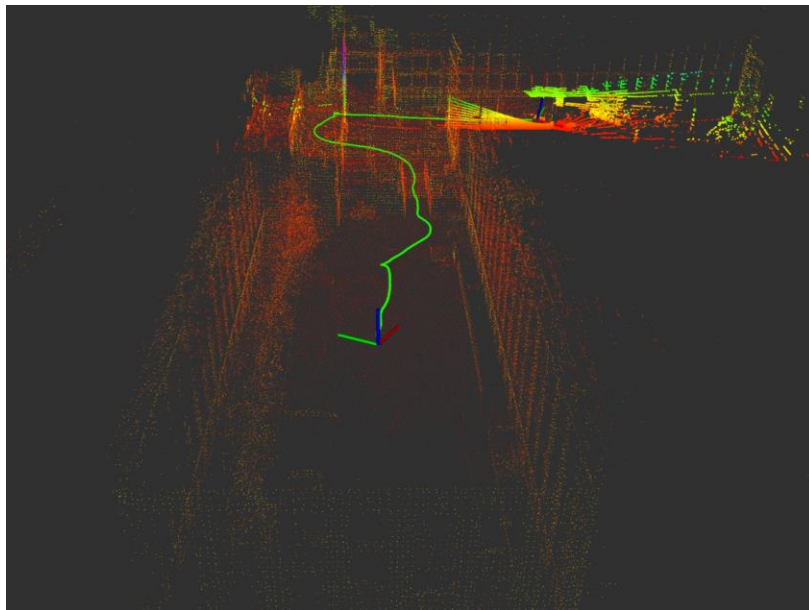
הינו אלגוריתם מבוסס גרף שמשלב מידע מ IMU ו Lidar ובעיקרו מותאם גרסת ה Ros 2 Foxy ובנוסף פועל בזמן אמת.

## שלבי הפעולה:

- הערכת תנועה בין סריקות LiDAR בעזרת נתוני IMU.
- התאמת נקודות מפתח (קצוות, משטחים) בין סריקות עוקבות.
- סינון פריימים – שמירה על פריימים אשר התקיים בהם שינוי של מקום או זמן.
- יצירת גרף מבוסס פריימים נבחרים עם זיהוי חזרה למיקום קודם (loop closure).

## יתרונות בולטים:

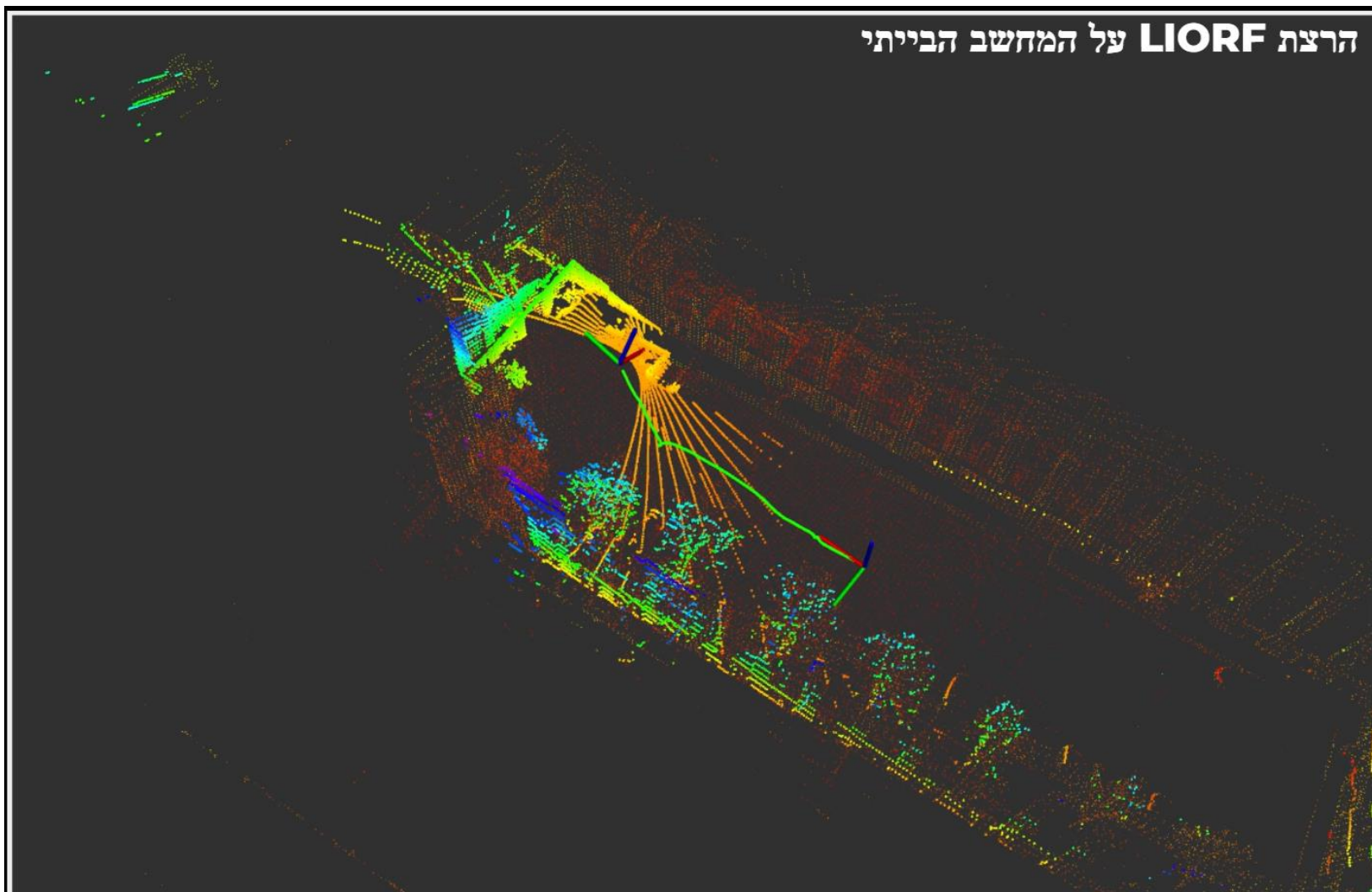
- מותאם לסביבות מורכבות בעלות שינויים מרחביים רבים.
- מפחית רעש חישובי בזכות סינון פריימים.
- תומך בחיישני Velodyne/Ouster ובמאגרי נתונים נפוצים.



תמונה מתוך הרצת הסימולציה של LIORF במחשב הביתי



# הדגמת סימולציות של LIORF



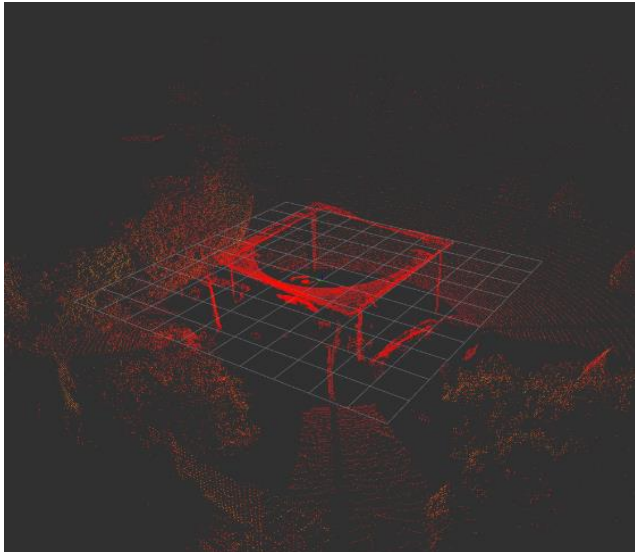
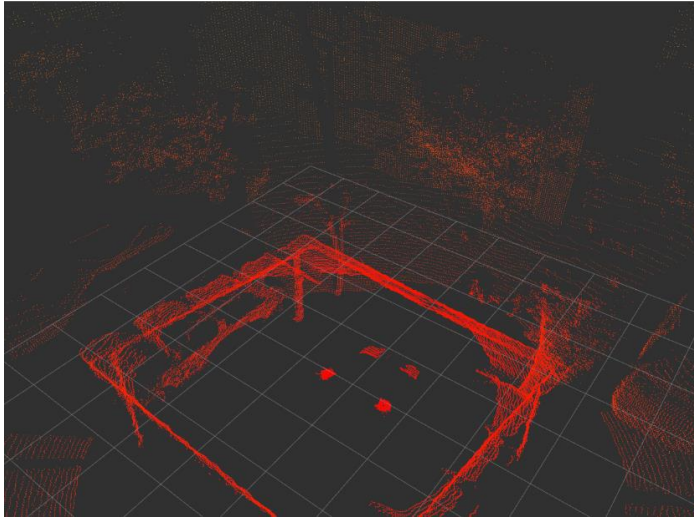
# תוצאות

## תוצאות סימולציה:

- הרצות מוצלחות עם מאגרי NCLT ו Kitti לאלגוריתמים LIOF ו FastLio בהתאמה תוך יצירת מפת SLAM תלת-מימדית תקינה הכוללת TF Tree תקין (קישור מלא בין כל הפריימים).

## תוצאות בזמן אמת ברכב:

- התקנת שני האלגוריתמים ללא תקלות במחשבי הרכב.
- אי הצלחה בהשלמת תהליך SLAM מלא לשני האלגוריתמים (לא נוצרה מפה דינאמית).
- אלגוריתם LIOF- הצליח לקלוט Point Cloud ולשדר ב RVIZ אך לא הושלמה בניית מפה עקב נתק ב TF Tree.
- אלגוריתם Fast Lio- תקלת חוסר תקשורת בין החיישנים לבין האלגוריתם – בעת ההרצה ברכב, נתוני ה Point Cloud של הליידר במוצא האלגוריתם לא נראו ב Rviz.



# דרישות כמותיות

אלגוריתם FAST LIO	אלגוריתם LIO SAM	
2.5-4.6Hz	10-12Hz	קצב היענות SLAM
לא נבחן בזמן אמת	לא נבחן בזמן אמת	דיוק מרחק סטטי
לא נבחן בזמן אמת	ללא נבחן בזמן אמת	דיוק מרחק דינאמי

# מסקנות

## הבנה עמוקה של מערכות SLAM

הפרויקט אפשר הבנה מעשית ותיאורטית של אלגוריתמים מתקדמים לשילוב Lidar ו IMU הכולל תהליכי סינון, התאמת סריקות ובניית גרף.

## חשיבות של סימולציה מוקדמת:

ההרצות הביתיות היו שלב קריטי לזיהוי כשלים פוטנציאליים מראש, שלב זה היווה הכנה מקדימה ומעמיקה אשר פרסה את התשתית לעבודה עם הרכב בשלב הבא

## שילוב בין תוכנה לחומרה במחשבי הרכב – האתגר המרכזי:

התאמה בין האלגוריתמים לגרסאות ROS עם החיבור לחיישנים אשר בעיות אלו בסיסן בתקלות תקשורת בין האלגוריתמים לחיישנים, הוכיחה את החשיבות של תיאום מלא לאורך כל המערכת.

## ערך מוסף לפרויקט הרכב האוטונומי:

גם ללא הצלחה מלאה בהרצת זמן אמת, הממצאים והתיעוד עם פתרון חלק מהתקלות מספקים תשתית להמשך פיתוח ולהמשך עבודה וחקירה ברכב האוניברסיטה.



# הצעות להמשך

## שדרוג סביבת העבודה

מעבר לגרסאות עדכניות של Ros2 ו ubuntu אשר יאפשר שימוש באלגוריתמים עדכניים יותר עם תמיכה טובה יותר בקוד פתוח.

## שימוש במאגרים מותאמים

המרה של מאגרי המידע באופן מלא לתמיכת Ros2 (כגון מאגרי ההקלטות מהרכב האוטונומי עצמו) או אפשרות של ביצוע פרויקט המרה מלא ממאגר Ros2 ל Kitty לצורך סימולציות אמינות.

## שיפור התקשורת בין החיישנים לאלגוריתמים

המשך חקירת בעיית ההתאמה בתקשורת בין החיישנים לאלגוריתמים (ופתרון בעיות קישור ה TF-Tree), אשר תוביל להמשך פיתוח מעמיק בנושא SLAM במחשבי הרכב.