המחלקה להנדסת תוכנה

פרויקט גמר – תשפ"ב

מציאת התאמה טובה ביותר בין עננים נקודות בעזרת רשתות גרפים עמוקות

Point Clouds Registration Based on SuperGlue Graph Neural Network

מאת

שם הסטודנט\ית: נתנאל ראובן ומאור בן יאיר

ת.ז סטודנט\ית: 312597784 205463938

מנחה אקדמי: דר' חסין יהודה **אישור:** **תאריך:**

מערכות ניהול הפרויקט:

#	מערכת	מיקום
1	מאגר קוד	GitHab
2	יומן	Monday
3	סרטון הסבר	Video

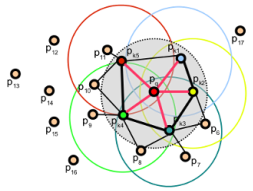
מידע נוסף:

סוג הפרויקט	**מחקרי ממרצה במכללה**
פרויקט מח"ר	**לא**
פרויקט ממשיך	**זה פרוייקט חדש**
פרויקט זוגי:	**כן**

תוכן עניינים –

מילון מושגים:

- **Point Cloud (ענן נקודות)** – מערך של נקודות במערכת קואורדינטות מסוימת. במערכת קואורדינטות תלת ממדית, נקודות אלה מסומנות בדרך כלל ב־Y, X ו־ Z.
- **Overlap (אחוז חפיפה אופטימלי)** – מבטא את ההתאמה האופטימלית אליה ניתן להגיע בין שני עננים נקודות כאחוז החפיפה בניהם.
- **התאמה חלקית** – בהנתן שני עננים נקודות A_{mx3} ו-B_{nx3} התאמה חלקית מתרחשת כאשר אחוז החפיפה בניהם לא שווה ל-Overlap. **
- **התאמה מלאה** – בהנתן שני עננים נקודות A_{mx3} ו-B_{nx3} התאמה מלאה מתרחשת כאשר אחוז החפיפה בניהם שווה לOverlap. **
- **Correspondence** – זוג נקודות מתאימות כך שעבור ההתאמה של עננים נקודות A_{mx3} ו-B_{nx3} הנקודות a_{jx3} ו-b_{ix3} נמצאות אחת על השניה.
- **Rotation Matrix (מטריצת סיבוב 3x3)** – היא מטריצת טרנספורמציה המשמשת לביצוע סיבוב במרחב האוקלידי.
- **Translation Vector (וקטור הזזה 3x1)** – משמש לביצוע הזזה במרחב האוקלידי.
- **SVD Singular value decomposition (פירוק לערכים סינגולריים)** - היא שיטת פירוק באלגברה ליניארית של מטריצה מרוכבת או ממשית. בהתאם לבעיה שלנו ניתן לחשב בהנתן זוגות של correspondence אופטימלים את מטריצת הסיבוב ווקטור הזזה האופטימלים עבור בעית התאמה ספציפית.
- **ICP (Iterative closest point)** – אלגוריתם איטרטיבי המשמש כדי למזער את ההבדל בין שני עננים נקודות. ICP משמש גם ל – 2D או 3D.
- **RANSAC (Random sample consensus)** – הוא טכניקת למידה להערכת פרמטרים של מודל על ידי דגימה אקראית של נתונים. בהינתן מערכים של עננים נקודות A ו-B כך שמכילים נקודות שיכולות להיות Correspondence וכאלו שלא, RANSAC משתמש בסכימת ההצבעה כדי למצוא ההתאמה הטובה ביותר.
- **Sinkhorn** – אלגוריתם הפותר את בעית ההשמה ע"י מציאת מינימום למחיר השגיאה (מרחק בין זוג נקודות), ובכך בבעיה שלנו ניתן למצוא correspondence set.
- **Descriptor** – תיאורים של המאפיינים החזותיים של התוכן לדוגמא בתמונות, סרטונים, אלגוריתמים או יישומים נוספים דומים לתיאורים אלו.
- **FPFH (Fast Point Feature Histograms) Descriptor** – תיאור גיאומטרי של נקודה בהתייחס לסביבה המקומית שלה.



- **Voxel down sample** – סינון נקודות על-ידי שימוש בייצוג של נקודה בודדת עבור גודל שטח מסוים.
- **Keypoint (נקודת מפתח)** – הן הנקודות החשובות ביותר שמסמנות את האוביקט (לדוגמא הקודקודים של ריבוע מסוים).
- **Dustbin** – משמש כ"פח" באלגוריתמים מאפשר לא להתייחס לחלק מסוים מהחישוב / דאטה.

**** נשים לב, שנקודות a ו-b מעננים נקודות A ו-B בהתאמה יכולות להיות חופפות ($||a-b||_2$ < threshold). אך לא תואמות (כלומר בפתרון האופטימלי אינן חופפות).**

מבוא

Lidar –

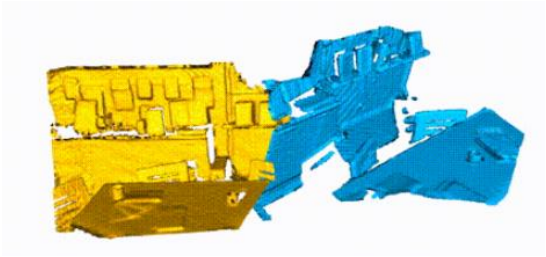
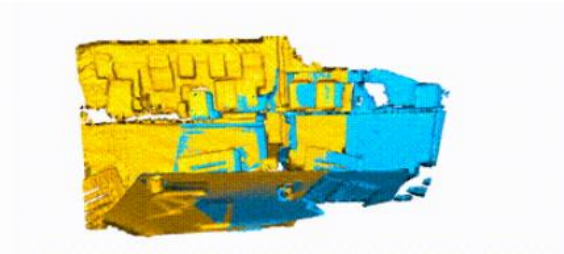
בשנים האחרונות בתחומים שונים בתעשייה נעשה שימוש במכשיר Lidar. זוהי טכנולוגיה למידול תלת מימדי / מדידת מרחק בעזרת חיישני לייזר. זאת על ידי הארת המטרה בקרן לייזר. הקרן נשברת וחוזרת אל סורק, אשר מחשב ע"י אורך גל הלייזר את מיקום המטרה. פלט הLidar הוא נקודות (x,y,z) במרחב ובעזרתו ניתן למפות את המרחב. דוגמא לשימוש ב-Lidar ניתן לראות בשימוש ברכבים אוטונומים, במפות 3D ועוד.

Deep Neural Network –

רשתות נוירונים עמוקות מהוות רכיב מרכזי בעולם התוכנה. רשתות נוירונים עמוקות הן מודל מתמטי חישובי, שפותח בהשראת תהליכים מוחיים או קוגניטיביים המתרחשים ברשת עצבית טבעית ומשמש במסגרת למידת מכונה.

SuperGlue – אלגוריתם שפורסם בשנת 2019 ע"י Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich. אלגוריתם אשר מבצע התאמה בין שתי תמונות. האלגוריתם מציע פתרון למציאת נקודות טובות ביותר להתאמה ודחיית נקודות שאינן ניתנות להתאמה. SuperGlue מורכב מרשת נוירונים גרפית (GNN) שלומדת להפיק Descriptor טוב ביותר לכל keypoint בכל אחת מהתמונות. לאחר מכאן מבצע התאמה בעזרת אלגוריתם Sinkhorn. (סקירת ספרות A3).

לאור הגדילה בשימוש הן ברשתות נוירונים עמוקות והן ברכיב הLidar.
בחרנו לבצע פרויקט מחקרי המשלב בין התחומים. **הבעיה שאיתה אנו נתמודד בפרויקט היא בעיית ההתאמה של שני עננים נקודות** (כאשר אחוז החפיפה בניהם 50=<). שהיא בעיה יסודית בתחום הנ"ל.

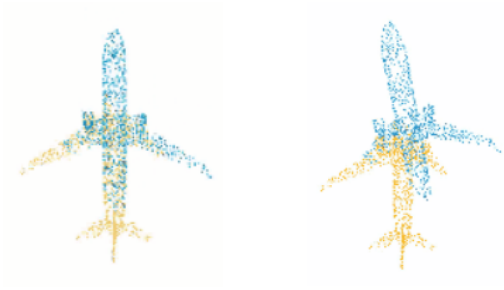



בפרויקט זה אנו נבצע התאמה לאלגוריתם SuperGlue. כך שיאפשר מציאת התאמה טובה ביותר עבור עננים נקודות. בעצם ילמד להפיק Descriptor טוב ביותר עבור כל נקודה מעננים הנקודות. לאחר מכאן יבצע התאמה בעזרת Sinkhorn.

דרישות ואפיון הבעיה

דרישות ואפיון הבעיה -

בעיית ההתאמה של שני עננ נקודות –
בהנתן עננ נקודות A_{mx3} ו-B_{nx3} אשר מייצגים את אותה **הסביבה** (או חלק זהה ממנה) מסובבות ומוזזות.
ואחוז חפיפה עבור ההתאמה האופטימלית Overlap.
נרצה למצוא מטריצה R_{3x3} ווקטור T_{1x3}, כך שהכפלת R והוספת T ל**כל נקודה** ב-A תוביל לכך שאחוז החפיפה של (R·A+T, B) יהיה קרוב ביותר שניתן ל-Overlap ובכך יספק לנו את ההתאמה הקרובה ביותר לאופטמלית.
ב-dataset שאיתו בחרנו לעבוד ETH (הסבר מפורט בהמשך) יש מטריצת סיבוב למיקום מוטעה R'_{3x3} ווקטור ההזה למיקום מוטעה T'_{1x3}, כך ש $A' = R'\cdot A+T'$.
לכן נרצה למצוא מטריצה R_{3x3} ווקטור T_{1x3} כך שאחוז החפיפה של (R·A'+T, B) קרוב ביותר שניתן ל-Overlap.



הבעיה מבחינת הנדסת תוכנה –

ראשית נבחין כי הבעיה שהצגנו מתחלקת לשתי בעיות האחת קלה והשניה קשה.

הבעיה הקלה: בהנתן וקטור זוגות של correspondence אופטימלים מצא את ההתאמה הטובה ביותר.
על ידי אלגוריתמם SVD ניתן להגיע למטריצת סיבוב ווקטור הזזה אופטימלים (כך שאחוז החפיפה לאחר התיקון שווה ל-Overlap).
הבעיה הקשה: מצא את וקטור זוגות הcorrespondence הטובים ביותר ובעזרתם מצא את ההתאמה הטובה ביותר. בהנתן שלא ידוע ה-correspondence מציאתן מהווה מרכיב מרכזי בבעיה. ככל שה-correspondence יהיו קרובים יותר לאופטימלים נוכל למצוא התאמה כך שאחוז החפיפה שלה קרוב יותר ל-Overlap.

כיום קיימים אלגוריתמים למציאת קירוב לפתרון הבעיה הקשה. לדוגמא, RANSAC + ICP לבעית ההתאמה או Sinkhorn למציאת correspondence ואז ע"י correspondence חישוב ההתאמה.

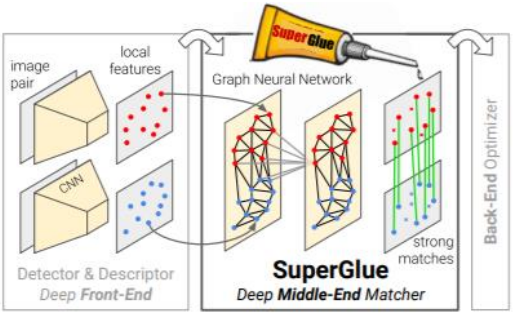
נרצה להשתמש באלגוריתם SuperGlue כדי למצוא פתרון קירוב טוב יותר לבעיה הצגנו. אופן פועלת רשתות הנוירונים עדיין נחקרות, ולכן אנחנו לא יודעים האם פתרון זה יהיה יעיל עבור בעיית עננ הנקודות.
כמה הרעש משפיע? מהם הפרמטרים שהרשת צריכה לקבל? איך הרשת לומדת? מה הרשת לומדת?
האם היא טובה על כל סוג של דאטה, גם כשאחוז החפיפה קטן?

סקירת בספרות

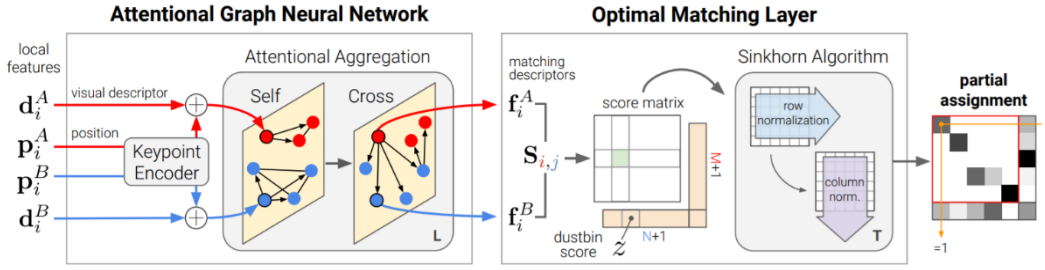
- (A1) Deep learning based point cloud registration
מאמר המציג גישות שונות המיושמות ברשות נוירונים לפתרון עבור התאמת שני עננים נקודות. מציג גישות שונות ואת היתרונות שלהן.

- (A2) RANSAC on FPFH Descriptors + ICP
מאמר המציג פתרון עבור התאמת שני עננים נקודות על ידי שימוש ב RANSAC אשר מבצע מה שנקרא Global Registration שזה בעצם התאמה ע"י מעט נקודות בדרך כלל, משיג התאמה טובה אך לא מדויקת. לאחר מכן מבצע ICP כמו שנקרא Local Registration שזה התאמה על ידי שימוש בכל הנקודות בעננים הנקודות (עובד אם ההזזה לא רחוקה מידי).

- (A3) SuperGlue
מאמר המציג פתרון עבור מציאת התאמה בין שתי תמונות. זהו האלגוריתם שאנו נתבסס על מנת לפתור את בעיית ההתאמה עבור עננים הנקודות.



SuperGlue פועל כ-middle-end שקושר קשרים בין keypoints, מבצע התאמה וסינון keypoints שלא עוזרים לבעית ההתאמה ובנוי בארכיטכטורת end-to-end.
הארכיטכטורה עובדת בשלושה שלבים:
שלב ראשון: ע"י שימוש ברשתות קונבולוציה מוצאת את המתארים המקומיים הטובים ביותר בכל אחת מן התמונות.
שלב שני: SuperGlue (פירוט בהמשך)
שלב שלישי: אופטימיזיר להערכת ההתאמה.



SuperGlue – כפי שניתן לראות SuperGlue מורכב משני רכיבים עיקריים:
הרכיב הראשון – Attentional Graph Neural Network המקבל וקטור P המיצג מיקום מסויים והDescriptor שלו D. Encoder פועל על ה-keypoint על מנת לחבר כל מיקום לכל Descriptor לוקטור יחיד. לאחר מכן משתמש בGraph Neural Network כדי ליצור ייצוגים חזקים יותר, f עבור כל, (p,d) כלומר Descriptors טובים יותר.
הרכיב השני – יוצר מטריצת מחיר S בגודל m_xn (כאשר m מספר הkeypoints בתמונה A ו-n בתמונה B). מוסיף למטריצה S dustbin ומוצא את ההתאמה החלקית האופטימלית ע"י Sinkhorn.

תיאור הפתרון

כמו שניתן לראות ממאמר (A1) ישנם גישות שונות לפתרון בעית ההתאמה על ידי רשתות נוירונים. אנחנו נתבסס עלמאמר SuperGlue (A3) המציע את הפתרון הטוב ביותר לבעית ההתאמה עבור תמונות נכון לשנת 2019. מאמר זה מציע פתרון ע"י שימוש ברשתות גרפים עמוקות למציאת הDescriptors הטובים ביותר. מאחר ו-SuperGlue התאימה את ה-keypoints בעזרת Descriptors ולא בעזרת שום דבר שמייחד את התמונה כ"תמונה". גרם לנו לתהות האם האלגוריתם הזה יכול לעבוד על עני נקודות? ואם כן, כמה טוב יצליח?

האלגוריתם יקבל שני עני נקודות ויעבוד בשני שלבים:

<u>בשלב הראשון:</u> רשת נוירונים גרפית שתאומן בכדי למצוא את הDescriptors טובים ביותר עבור כל נקודה בכל אחד מעני הנקודות.

<u>בשלב השני:</u> Sinkhorn שיאמד את מחיר השגיאה הנמוך ביותר עבור כל זוג נקודות על פי הDescriptors שלהם. ובכך ימצא correspondence set טוב.

כפי שהזכרנו אלגוריתם SuperGlue חדשני ומספק תוצאות מעולות, אין שום סיבה שלא נבחן את יכולתו לעבוד על Data שונה (עני נקודות).
נשאף להגיע לרמת דיוק גבוהה מהקיים היום. כמו כן להבין מהם הערכים המשמעותים בכל אחד מהשלבים של האלגוריתם, מהם הערכים אשר יובלו את הרשת להתכנס לפתרון מציאת ההתאמה הטובה ביותר ואיך היא בנויה.

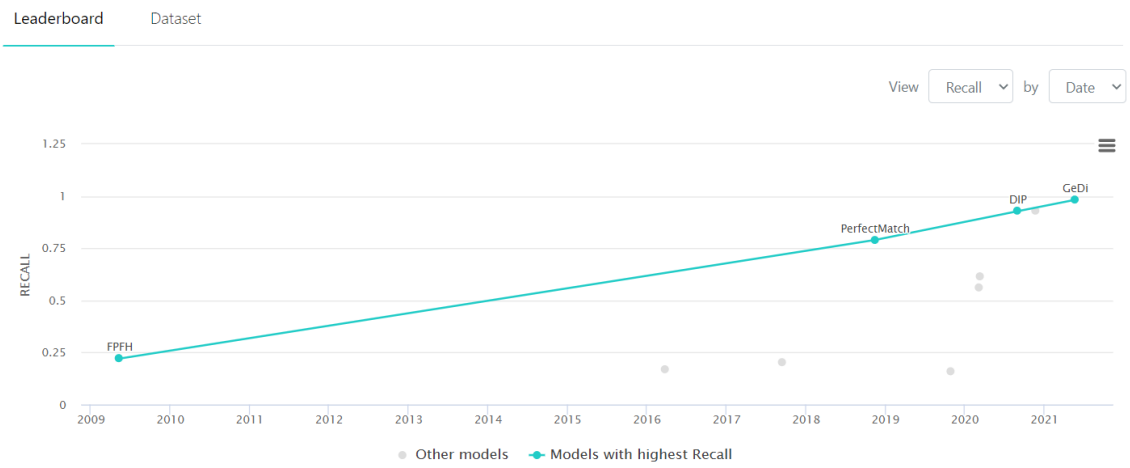
לשם השוואה ראשונית השתמשנו בארבעה אלגוריתמים המציעים פתרון לבעיית ההתאמה. בחנו את האלגוריתמים המצוינים מטה על עשר בעיות מכל קבוצה בדאטה ETH ובכך הרצנו כל אחד מהאלגוריתמים על אותן 80 בעיות. לקבצות הבעיות הזו קראנו POC.
אלגוריתמים ראשוניים לבדיקת הבעיה:

1. ICP
2. RANSAC + ICP
3. Sinkhorn + SVD
4. Sinkhorn + RANSAC + ICP

למעשה כל שנרצה לעשות זה למצוא התאמה טובה יותר מאלגוריתמים אלו כצעד ראשוני. כמו כן, נרצה להיות סקלבילים בבחירת הדאטה, ולשם כך בחרנו דאטה המצייג סביבות שונות שמודלו (הסבר במה עשינו עד כה).

כאמור, נשים לב כאלגוריתמים שהצגנו מקודם אינם מהווים את הציונים הטובים ביותר עבוד בעיית ההתאמה על הדאטה של ETH וישנם אלגוריתמים שפותרים את הבעיה הזו בציונים טובים בהרבה. נרצה להשוות את יכולות האלגוריתם שנבנה ליכולתיהם.

Point Cloud Registration on ETH (trained on 3DMatch)



ציוני האלגוריתמים הטובים ביותר עבור בעיית ההתאמה ב-dataset ETH.

מטרת העל שלנו בפרויקט תהיה לעבור את הציונים של האלגוריתמים כמו GeDi ו-DIP עבור בעית ההתאמה של שני עננים נקודות על dataset ETH. בהמשך נבחן גם על datasets נוספים.

קדם מחקר (מה עשינו עד כה)

כמו שצינו בתיאור הפתרון בחנו את האלגוריתמים בבחינה ראשונה על אותן 80 בעיות התאמה. בכל הרצה כיוונו את הפרמטרים בכדי להגיע לציונים טובים ככל שניתן על קבוצת הPOC. את ההרצות בצענו על המחשבים האישים שלנו ולכן היו מעט הגבולות. (בהמשך נריץ את אלגוריתמים אלו על מחשב יעודי, ובכך נוכל אף להגיע לתוצאות טובות יותר).

הכנת Data (לינק לאתר הדאטה) –

טבלה הסבר על הדאטה (5)

Data set names	Situation	Environments	Dynamics	Avg. points
Apartment	Indoors	Structured	Furnitures moved in between scans	365,000
ETH Hauptgebaude	Indoors	Structured with repetitive elements	Walking persons	191,000
Stairs	Mixed	Structured with large variations in scanned volumes	None	191,000
Mountain plain	Outdoors	Semi-structured with low vertical constrains	None	102,000
Gazebo in summer	Outdoors	Semi-structured (sparse vegetation)	Seasonal changes (see winter)	170,000
Gazebo in winter	Outdoors	Semi-structured (sparse vegetation)	Seasonal changes (see summer)	153,000
Wood in summer	Outdoors	Unstructured (dense vegetation)	Seasonal changes (see autumn)	182,000
Wood in autumn	Outdoors	Unstructured (dense vegetation)	Seasonal changes (see summer)	178,000

עבור התאמת עננים הנקודות השתמשנו ב-ETH Dataset. מכיל שמונה סוגי סביבות שונות המיוצגות על ידי עננים נקודות להתאמה. ובעל כ-800 בעיות התאמה שונות. דוגמאת לאזורים שנסרקו בנספחים A.

כמו כן מכיל קובץ המכיל:

1. שמות של זוגות עננים הנקודות עליהם נבצע התאמה (מקור ויעד).
2. T - מטריצת הטרנספורמציה ליצירת בעיית ההתאמה.
3. Overlap – ערך המייצג אחוז חפיפה כאשר ההתאמה הינה אופטימלית בין עננים הנקודות.

על-ידי מטריצה T הזזנו וסבבנו את ענן נקודות המקור. הפעלנו אלגורתמים שונים שיתקנו את מיקום המקור בשאיפה להגיע לאחוז חפיפה (בין המקור המתוקן והיעד) קרוב ביותר שניתן ל-Overlap.

כל ענן נקודות מכיל מספר נקודות כפי שמצוין בטבלה (5) שכל נקודה מייצגת (x,y,z) במרחב וביחד מייצגים את הסביבה שנדגמה.

מדדים להצלחה –

על מנת לבחון מהו הציון עבור כל התאמה. נסמן אחוז החפיפה שהתקבל מתוצאת ההתאמה ב Score בהשוואה ל-overlap. ההשוואה תתבצעה בצורה הבאה.

- אם Overlap > Score הציון הינו $\frac{\text{Score}}{Overlap}$.
- במידה ו-Score < Overlap הציון הינו $\frac{\text{Score}}{Overlap}$ - 2.

ישנם מקרים בהם אחוז החפיפה של תוצאת ההתאמה (Score) גבוה מהOverlap מאחר ובמיקום המוטעה מספר הנקודות החופפות גדול, אך בתוצאה האופטימלית הן אינן חופפות, במידה ודבר זה קורה "נעניש".

**** מדד זה משמש אותנו כמדד ראשוני בלבד. בהמשך נבדוק מדדים שונים כמו המרחק ממצטרית הסיבוב ווקטור ההזזה האופטימלים.**

לשם מימוש ובדיקה של האלגורתמים הראשונים שבדקנו השתמשנו בספריות:

- Open3d – ספריה המתמקדת ב3D דאטה. מספקת כלים כמו ויזואליזציה, מבנה נתונים ואלגוריתמים עבור סוג דאטה זה.
- POT – ספריה שמקנה כלים לפתרון בעיות אופטימזציה (לדוגמא Sinkhorn).

ICP –

על-ידי שימוש בספריית Open3D השתמשנו באלגוריתם ICP על כל זוג של ענני נקודות. כאמור ICP טוב לבעיות בהן האתחול (המיקום הראשוני של ענני הנקודות) לא משמעותי. במידה וענני הנקודות רחוקים אחד מהשני (הן בסיבוב או בהזזה), סביר להניח שהאלגוריתם לא יתכנס או שישנה את המיקום במעט אבל עדיין יהיה רחוק מתוצאת ההתאמה הרצויה. וזה אכן מה שקרה אצלנו.
המיקום ההתחלתי של המקור ב-ETH גם מסובב וגם מוזז מה שמקשה על ICP להתכנס מאחר ומתייחס לכל הנקודות בענני הנקודות (בין 90,000-370,000 נקודות). תוצאות על הPOC עמדו על ~ 5%

RANSAC + ICP (A2) (לינק לקוד) –

על-ידי שימוש בספריית Open3D השתמשנו באלגוריתמים RANSAC ו-ICP על כל זוג של עננים נקודות. בשלבים הבאים:

1. בצענו Voxel down sample עבור כל ענן נקודות. בכדי לצמצם את מספר הנקודות בשטח מסוים (לדוגמה - במקום 50 נקודות ב10 סמ"ר נסתכל על נקודה אחת בכל 10 סמ"ר).
2. חשבנו את FPFH Descriptors עבור נקודה מכל אחד מעננים הנקודות – מבטא את הקשר של הסביבה הלוקאלית של כל נקודה.
3. הרצנו RANSAC על ה-FPFH Descriptors לקבלת מטריצת טרנספורמציה. RANSAC מקרב את המקור ליעד בהתייחסות למעט נקודות ולכן לא מניבה תוצאת התאמה טובה אך התוצאה מספיק טובה כדי ש-ICP יוכל להתכנס ולקרב את ההתאמה להתאמה טובה.
4. הרצנו ICP כאשר השתמשנו במצטריצת התוצאה של שלב 3 כמיקום התחלתי עבור ICP. ומאחר וההתאמה יחסית קלה ההתאמה השתפרה משמעותית.

תוצאות על POC:
הציון הכללי על כל קבוצת ה-POC היה 88%.

```
wood_autumn 's score:  0.8063904762195285
gazebo_summer 's score:  0.927800988197409
gazebo_winter 's score:  0.9172904821848462
wood_summer 's score:  0.7881205522373488
stairs 's score:  0.9110179256426726
apartment 's score:  0.9472178433426762
hauptgebaude 's score:  0.833517106661834
plain 's score:  0.9161302839790727

total avarage =  0.8809357073081735
```

Sinkhorn + RANSAC + ICP (לינק לקוד) –

על-ידי שימוש בספריות Open3D ו-POT, השתמשנו באלגוריתם Sinkhorn RANSAC ו-ICP.
בשלבים הבאים:

1. עבור כל זוג עננים נקודות אתחלנו וקטורים a(source) ו-b(target) המייצגים את משקל כל אחת מנקודות הענן (מחושב על פי dustbin weight–1 חלקי מספר הנקודות בענן).
2. בצענו Voxel down sample עבור כל אחד מעננים נקודות. על מנת לצמצם את מספר הנקודות בשטח מסוים.
3. חשבנו את ה-FPFH Descriptors עבור כל אחד מעננים הנקודות בכדי לחשב את מטריצת המחיר M (מחשבת מרחק בין נקודה במקור ליעד).
4. על-ידי המשקלי הנקודות a ו-b ועל-ידי מטריצת המחיר M אנו מריצים את Sinkhorn בכדי למצוא Correspondence set טוב שממנו נרצה להריץ התאמת נקודות.
5. עבור מטריצת המשקלים M הוספנו עמודה ושורה של אפסים. ולוקטורים a ו-b הוספנו ערך נוסף בסוף הוקטור שהוא dustbin weight. ובכך עבור נקודות שאין התאמה טובה לא נתאים.

6. לאחר מכן הרצנו RANSAC על הCorrespondence set שמצאנו כדי לקבל את מטריצת טרנספורמציה שמקרבת את המקור ליעד בהתייחסות למעט נקודות.
7. הרצנו ICP – כאשר השתמשנו בתוצאת RANSAC כאתחול לICP .

תוצאות על POC:

```
wood_autumn 's score:  0.9612706949122843
gazebo_summer 's score:  0.8379812701900742
gazebo_winter 's score:  0.9764072870882348
wood_summer 's score:  0.8721462220092204
stairs 's score:  0.8254879773663045
apartment 's score:  0.7041723101265693
hauptgebaude 's score:  0.7521560178754433
plain 's score:  0.7748627023139193
total avarage =  0.8380605602352563
```

הציון הכללי על כל קבוצת ה-POC היה 84%. **

Sinkhorn + SVD (לינק לקוד) –

על-ידי שימוש בספריות Open3D ו-POT, השתמשנו באלגוריתימים Sinkhorn ו-SVD על מנת למצוא נקודות חשובות ביותר, כרדוקציה לבעיית ההשמה (בעית ההשמה הינה מציאת התאמה חד-חד ערכית לכל נקודה, כך שהמחיר לבחירת נקודות תואמות הוא מינימלי).

בשלבים הבאים:

שלבים 1-5 פועלים באותו אופן ש-sinkhorn + RANSAC + ICP עובד.

בשלב השישי: השתמשנו באלגוריתם SVD על מנת לקבל את מטריצת הסיבוב ווקטור ההזזה.
כלומר החלפנו את RANSAC + ICP ב-SVD.

תוצאות: בתהליך.. **

**** כאמור בכל האלגוריתמים שבדקנו השתמשנו בseed קבוע.**
**** נבחין כי Sinkhorn נבחן עם voxel גבוה יותר משאר האלגוריתמים (עקב מגבלות מקום בשל יצירת מטריצות גדולות), ובכך התייחס למספר קטן יותר של נקודות. בהמשך נבחן על מחשב יעודי בצפיה להגיע לתוצאות טובות יותר.**

נספחים

תכנון הפרויקט

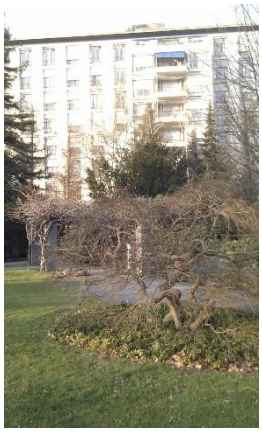
17.10.21	סיום למידה על רשתות נוירונים עמוקות
12.11.21	בחירת נושא – cloud view matching in Lidar והתחלת מחקר
07.12.21	סיום למידה על אלגוריתמים של Lidar
09.01.22	בניית References שישמשו כציון בסיס איתו נרצה לעבור
28.02.22	בניית שלד של הרעיון
31.03.22	כיוונון פרמטרים לאלגוריתם
31.04.22	השוואת התוצאות לתוצאות האלגוריתמים הבסיס שיצרנו

A. Data -

1. Apartment – תמונות של הדירה שנדגמה –


2. Gazebo Summer / Gazebo Winter – תמונה של העץ שנגדם –


בקיץ


בחורף

3. Hauptgebaude – תמונה של המבנה הנדגם


B. דוגמאות לבעיית ההתאמה ודוגמא של אלגוריתמים כגון ICP / Sinkhorn / RANSAC –

1. דוגמאות לתוצאות רצויות

2. דוגמאות לבעיות

3. דוגמאות לתוצאות RANSAC + ICP

4. דוגמאות לתוצאות Sinkhorn + RANSAC + ICP

5. דוגמאות לתוצאות Sinkhorn + SVD