

מגשים:

ליאון גורין --- תז 214511214

איתי בר --- תז 325839710

## תרגיל בית 1 בראייה ממוחשבת

### פנורמות מישוריות

תיאור האלגוריתם:

עבור כל פאזל:

1. טוענים את כל החתיכות ומייצרים עותקים שלהם בדרגות אפור.
2. מייצרים את התמונה הסופית target כמטריצה מלאה באפסים בגודל הנתון.
3. מפעילים את הטרנספורמציה הנתונה על החתיכה הראשונה.
4. מעתיקים את החתיכה אל ה target.
5. כל עוד אנחנו מצליחים לחבר חתיכות ולא הצלחנו לפתור את כל הפאזל:
6. עבור כל חתיכה לא פתורה:
  6. מחשבים פיצרים של SIFT עבור החתיכה ועבור תמונת ה target.
  7. מחשבים מטריצת שכנויות בין כל זוגות הפיצרים.
  8. מבצעים התאמה בין כל פיצר לפיצר שהכי דומה לו.
  9. מפלטים פיצרים שיש להם יותר מפיצר אחד שדומה להם.
  10. אם מספר הפיצרים שמצאנו קטן מהיפרפרמטר שקובע מספר פיצרים דרוש מינימלי, אזי לא הצלחנו לחבר את החתיכה כראוי ונמשיך לחתיכה הבאה.
  10. מבצעים RANSAC למשך מספר איטרציות קבוע למציאת הטרנספורמציה הטובה ביותר בין החתיכה לתמונת ה target. אנחנו השתמשנו בהרחבה הנקראת MSAC:
  11. בוחרים רנדומלית 3 או 4 זוגות פיצרים תואמים (בהתאם לסוג הטרנספורמציה, אפינית או הומוגרפית).
  12. מתוך הנקודות הללו מוצאים את הטרנספורמציה בין החתיכה ל target.
  13. מפעילים את הטרנספורמציה שמצאנו על כל הפיצרים של החתיכה.
  14. עבור כל פיצר שהפעלנו עליו את הטרנספורמציה, מחשבים את המרחק האוקלידי בינו לבין הפיצר התואם לו ב target.
  15. מחשבים פונקציית מחיר M-estimation מתוך המרחקים האלו המשערכת באופן טוב יותר את התאמת החתיכה.
  17. אם המחיר קטן מהמחיר הכי קטן שמצאנו עד כה, נשמור אותו ואת הטרנספורמציה שמצאנו.
  18. אם המחיר הכי טוב שמצאנו חלקי מספר הפיצרים התואמים נמצא מעל לסף מסוים, אזי לא הצלחנו לחבר את החתיכה כראוי, ונמשיך לחתיכה הבאה.
  19. נפעיל את הטרנספורמציה הטובה ביותר שמצאנו על החתיכה.
  20. נעתיק את החתיכה בצבע והחתיכה האפורה לתמונת ה target הצבעונית והאפורה.
  21. נסמן את החתיכה כפתורה.
  22. נמשיך לחתיכה הבאה.
  23. כאשר סיימנו לחבר את כל החתיכות, או שהייתה איטרציה שבה עברנו על כל החתיכות ולא הצלחנו לחבר אף אחת מהן, סיימנו מעשית את הפאזל.
  24. נשמור את התוצאות בקבצים על פי הדרישות.

### שימוש בהרחבה של RANSAC הנקראת MSAC

לאחר מספר ניסיונות הגענו למסקנה שהאלגוריתם של RANSAC לא מספיק רובאסטי עבור ההרצות שלנו. זה הוביל אותנו לחפש אלטרנטיבות יותר טובות, ולבסוף החלטנו ללכת על MSAC שהיה יחסית פשוט למימוש. MSAC להבנתנו מנסה למקסם את ה likelihood במקום רק את מספר ה inliers. מניסויים שערכנו, קיבלנו תוצאות טובות יותר עם MSAC.

### חיפוש היפרפרמטרים

בחירה מיוחדת במימוש שלנו היא התאמה של ההיפרפרמטרים הרבים באלגוריתם שלנו עבור כל פאזל במיוחד. לשם כך השתמשנו בסקריפט חדש שמשמש בספריית Optuna על מנת לבצע חיפוש היפרפרמטרים באופן אוטומטי ובאמצעות אלגוריתמים מתקדמים. הגדרנו שהתוצאה שאנחנו רוצים לאפסם היא מספר החלקים שהצלחנו להרכיב בפאזל, ולכן אנחנו מבקשים למקסם את היחס בין מספר החלקים שהצלחנו לחבר למספר החלקים הכולל. עבור כל פאזל, הרצנו את האלגוריתם החיפוש למשך 100 איטרציות עם היפרפרמטרים שנבחרו אוטומטית על ידי הספרייה. בסיום ההרצות אנחנו מקבלים תוצאה של כל ההרצות שהתקיימו יחד עם גרפים שימושיים, ומשם לקחנו את ההיפרפרמטרים הכי מוצלחים עבור הפאזל הספציפי. מצאנו שהשימוש באלגוריתם חיפוש יעיל את תהליך העבודה ושיפר את התוצאות שלנו. הוא למעשה חסך לנו את הצורך בחיפוש ידני של ההיפרפרמטרים.

### שיפורי זמן ריצה במהלך הרצות שלנו

בחירה נוספת שעשינו הייתה להשתמש במימוש של scipy בחישוב מטריצת השכנויות במהלך הניסויים שלנו על מנת לשפר את זמן ההרצה, וזאת מפני שמצאנו שהחישוב הספציפי הזה לקח את החלק הכי גדול מבחינת זמן הריצה. המימוש של scipy מהיר יותר באופן משמעותי. לצורך הגשת התרגיל השתמשנו במימוש יעיל יחסית משלנו שמשמש ב numpy. בנוסף, השתמשנו בספריית numba על מנת להאיץ חלק מהפונקציות האחרות, כאשר השתמשנו בפונקציות אחרות יעילות פחות לצורך ההגשה.

כמו כן, השתמשנו ב multithreading על מנת להריץ במקביל את כל הפאזלים ולקבל תוצאות מהר יותר. גם כאן החלפנו לפונקציה שפותרת את כל הפאזלים אחד אחרי השני לצורך ההגשה.

### איכות התוצאות, הקשיים והכוונות לשיפור

אנחנו חושבים שהגענו לתוצאות טובות יחסית לאיכות הפאזלים שניתנה, רוב הפאזלים ההתחלתיים שהיו "קלים" יותר הצלחנו לפתור להם את כל החלקים ובפאזלים "קשים" יותר הקפדנו למצוא התאמות הגיוניות ככה שיהיו כמה שפחות חתיכות "מרוחות" על המסך (ככל האפשר). נציין שעם עוד זמן, איכות התוצאות שלנו יכלה רק לעלות כי בשלב מאוחר, אחרי שמימשנו את כל השיפורי האלגוריתם שיכולנו להמציא, חיבור התמונות נהפך למשחק "חפש את הפרמטר" שהשקענו לא מעט שעות בשביל לנסות להשיג תוצאות מספקות. אנחנו חושבים שהקשיים העיקריים של הפאזלים ה"קשים" נבעו ממחסור בפיצ'רים ש SIFT מצא בתמונות הניתנות שזה יכל להיות מהמון סיבות כמו, איכות, עיוות לא טריוויאלי וכדומה.

כדי לקבל תוצאות יותר טובות יכולנו לשפר את לולאה ה RANSAC, לחפש הרכבות יותר טובות כמו MLESAC, יכולנו להוסיף היפרפרמטרים לחישוב וכמובן יכולנו להשקיע עוד זמן לחפש את הפרמטרים המושלמים עבור כל פאזל.

שיפור נוסף שהיינו יכולים לנסות הוא שבמקום שבכל איטרציה על החתיכות נחבר את כל החתיכות שפונקציית המחיר עבורן קטן מסף מסויים, אנחנו יכולים לחבר רק את החתיכה שעבורה קיבלנו את פונקציית המחיר הקטנה ביותר ולהתחיל איטרציה חדשה על כלל החתיכות שנותרו. זה עשוי לשפר את התוצאות מכיוון שקודם אנחנו נחבר את החתיכות שאנחנו הכי בטוחים לגביהן (ונמנע מחתיכות שאולי כן ניתן לחבר אותן בשלב זה אך באופן לא הכי מדוייק). חתיכות אלו יהוו נקודת התחלה טובה יותר על מנת לחבר חתיכות נוספות. עם זאת, גישה זו עלולה להוסיף זמן ריצה משמעותי.

דבר נוסף שקיבלנו בתוצאות שלנו הוא קווים שחורים המפרידים בין החתיכות בתמונת הפאזל השלמה. בתחילה הנחנו שזאת גישת העבודה על מנת לזהות בקלות הפרדה בין החתיכות. אמנם לאחר מחשב מחודשת, יתכן ותוצאה זו לא רצויה ואף פוגמת באיכות התוצאה מכיוון שאלגוריתם ה SIFT עשוי לזהות את הקווים השחורים כאזורים בעלי רגישות גבוהה ל Difference of Gaussians, אזורים אלו הם למעשה edges, ובכך לקבוע את נקודות המפגש בין הקווים כפיצ'רים לגיטימיים של התמונה. לכן, כדאי לנסות למצוא דרך להעלים את הקווים השחורים האלו. דרך אפשרית היא באמצעות blinding בהרכבה של החתיכות.