

# ראייה ממוחשבת – תשפ"ב

## תרגיל בית 1 – פנורמות מישוריות

31/3/2023

### הנחיות כלליות:

- ההגשה היא בזוגות או ביחידים בלבד.
- בתרגיל זה יש לממש כל שאלה בשפת מטלב או פייתון, כאשר ניתן להשתמש בספריות קיימות לפעולות בסיסיות כמו קריאה ושמירה של תמונות, ציור על גבי תמונה ופונקציות יותר ספציפיות שמותר להשתמש בהן לפי הפירוט בכל שאלה. ספריות מומלצות במטלב הן: Image-Processing and Computer-Vision Toolboxes כמו גם ספריית vl\_feat. עבור פייתון אפשר לציין את Pillow, OpenCV, scikit-image.
- יש להגיש את כל הקוד שאתם כתבתם כתבתם (לא כולל קוד מספריות אחרות שנעזרתם בהן) בקובץ יחיד בשם code.zip.
- יש להגיש את כל התוצאות שתבקשו להציג כאשר יש תיקייה נפרדת לכל פאזל עם שם הפאזל, כשהכל ארוז בקובץ יחיד בשם results.zip.
- יש להגיש מסמך אחד בפורמט pdf, עם כל התוצרים וההסברים הנלווים לכל שאלה, שבו יש לציין שם ותעודת זהות.
- את התרגיל יש להגיש דרך אתר המודל. אפשר להגיש עדכונים של ההגשה (כל עוד זה לפני מועד ההגשה) כמו גם ערעורים לבדיקה במקרה הצורך במייל חוזר באותו שרשור.
- התרגיל להגשה עד ליום שני ה-24 באפריל בשעה 23:59. כל יום איחור יגרור הורדה של 4 נק', עד לתאריך אחרון של יום חמישי ה-27 באפריל. לא תינתן דחייה כלשהי (פרט למקרים מאושרים ספציפיים).

### שאלות:

#### 1. (50 נק') Hough Transform on ellipses

- המטרה היא לפתור (עד כמה שאפשר!) מספר פאזלים מישוריים. בכל פאזל יש מספר מסוים של חלקים שצורתם ריבוע, אשר מתחברים לאחר פעולות קצוץ שאינן נתונות לפסיפס אחד אם חפיפות בין החלקים. הפאזלים השונים נבדלים בפרמטרים הבאים (מה שגם משפיע על דרגות הקושי):
- א. מספר החלקים הכללי
  - ב. סוג הטרנספורמציה (warping) האפשרי: affine/homography
  - ג. רמת העיוות של הטרנספורמציות (עד כמה המרובע המתקבל רחוק מלהיות ריבוע באותו הגודל)
  - ד. עיוותים פוטומטריים של התמונות (כך שפיקסלים מתאימים בין תמונות אינם בהכרח מאותו הצבע)
- ראו דוגמא מפורטת בדף הבא לפאזל והפתרון המושלם שלו

# דוגמא לפאזל

11 תמונות קלט (הסדר הוא אקראי)

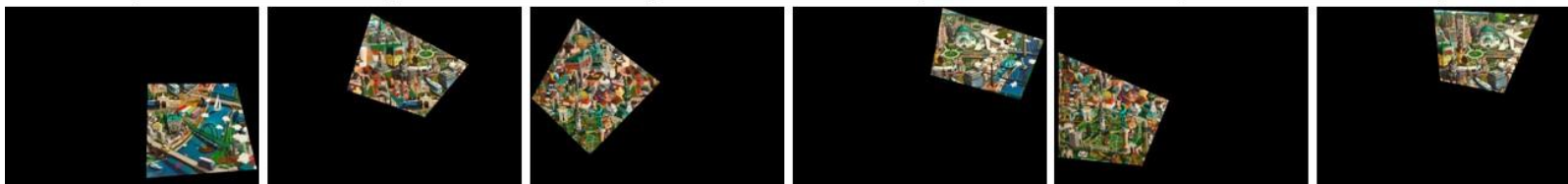


פתרון פנורמה מישורית (כזה תצטרכו לייצר)



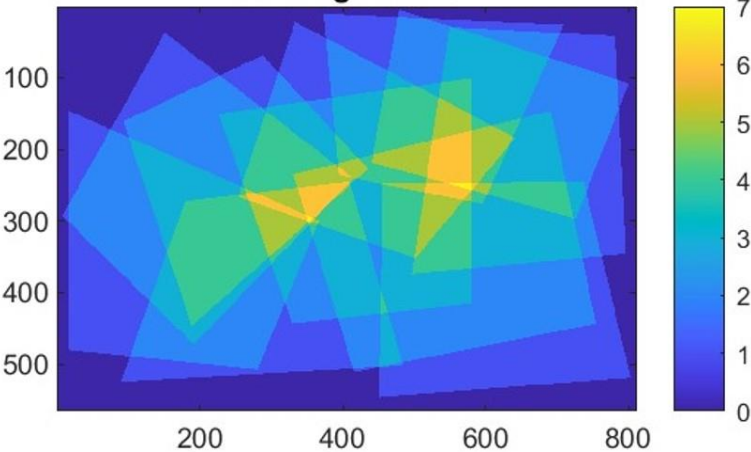


המיקומים היחסיים של החלקים:

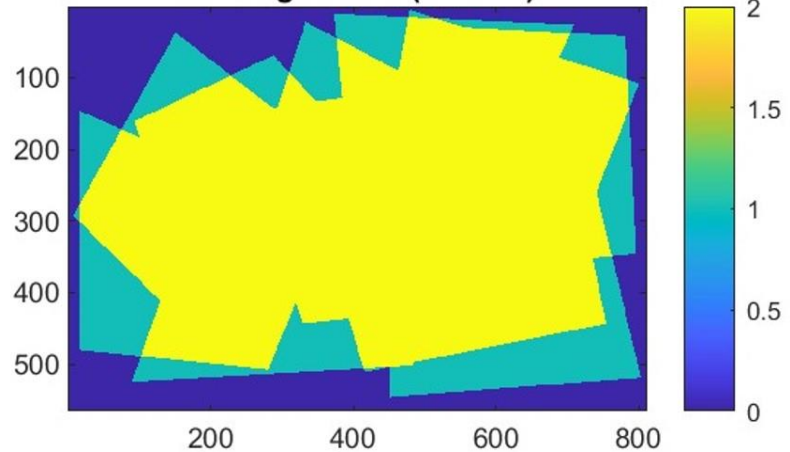


מפות שמראות את כיסוי השטח שהתקבל:

coverage count



coverage count (0 1 or 2)



קלטים לכל פאזל:

א. תיקיית pieces שבה החלקים הריבועיים של הפאזל

ב. הטרנספורמציה שמביאה את piece\_1 למרחב שבו צריך לשבת בסוף הפתרון הסופי (ינתנו גם מימדי המרחב). זה לא אומר שאתם צריכים להתחיל את הפתרון מהחלק הראשון, אלא זה אמור לתת לכם את העוגן הסופי כדי לשים את הפאזל שהרכבתם באוריינטציה המקורית. לדוגמא, יכול להיות שחיברתם את כל החלקים בצורה טובה ויצא לכם תוצאה שהיא באופן גלובלי מסובבת, עם aspect-ratio מעוות וכו'. הנתון של הטרנספורמציה יעזור לכם ליישר את כל הפאזל לפרספקטיבה הנכונה ולייצר תוצאה שנראית כמו בדוגמא (עם רקע שחור איפה שאין חלקים בפתרון). הטרנספורמציה נתונה בתוך הקובץ מהשם warp\_mat\_1\_\_H\_XXX\_\_W\_YYY\_.txt כאשר XXX ו YYY הם גובה ורוחב התמונה השלמה, בהתאמה.

שלבים לפתרון:

א. מצאו מימוש קיים של SIFT (מימוש של detection ו- description ולא יותר מזה) והפעילו אותו על כל אחת מהתמונות בפאזל. כזכור, תוצאת ה-detector, לכל נקודת עניין (IP), הוא וקטור  $[x, y, r, t]$ , כאשר  $x$  ו- $y$  הם קואורדינטות מיקום ו- $r$  ו- $t$  הם ה-scale וה-orientation, ותוצאת ה-descriptor הוא וקטור 128 ממדי.

- ב. עבור זוגות של תמונות (או זוגות של אזורים מורכבים של הפאזל) שיש בהן  $M \times N$  נקודות עניין בהתאמה, ממשן את חישוב מטריצת המרחקים מגודל  $N \times M$  בין כל זוגות וקטורי ה-descriptors.
- ג. חלצו את ההתאמות (matches) שעברו את מבחן ה ratio-test
- ד. מצאו מימושים קיימים של solver מינימלי - כזה שמשתמש ב-3 התאמות להתאמת טרנספורמציה אפינית וכזה שמשתמש ב-4 התאמות להתאמת הומוגרפיה. הכוונה אך ורק ל solver שהקלט שלו זה רק הקואורדינטות של 3 (או 4) זוגות הנקודות (לא פונקציה שמקבלת תמונות, או את כל ההתאמות).
- ה. ממשן את ההפעלה של הטרנספורמציה על סט נקודות ובעזרת זה את חישוב ה- residuals ביחס לטרנספורמציה.
- ו. ממשן לולאת RANSAC פשוטה שמשתמשת ב- solver ובחישוב ה-residuals. שכללו את ה-RANSAC כרצונכם.
- ז. מצאו מימוש קיים של פונקציית warping, אשר יכולה להפעיל את הטרנספורמציה על תמונה (רצוי לבחור אופציה מתקדמת של אינטרפולציה, כגון bilinear או bi-cubic).
- ח. עבור זוגות של תמונות (או זוגות של אזורים מורכבים של הפאזל) העתיקו באמצעות warping את תמונת ה-source לתמונת ה-target, על בסיס הטרנספורמציה שהותאמה, על מנת לחבר בין חלקים או אזורים של הפאזל.

#### הערות:

- א. רצוי להעביר את התמונות ל-grayscale וניתן לשנות את הגודל שלהן.
- ב. ניתן לשחק עם thresholds ופרמטרים לכל אורך התהליך. אפשר גם לבחור פרמטרים שונים עבור תמונות קלט שונות.
- ג. בכל פאזל מופיע השם affine או homography, מה שמעיד על סוג הטרנספורמציה הנדרשת.
- ד. שימו לב שחלק מהקלטים קשים. יתכן שלא ניתן להגיע לתוצאות מושלמות, אלא רק לנסות לפתור ולדייק במידת האפשר.
- ה. הקוד חייב לעבוד אוטומטית מההתחלה עד הסוף, בלי התערבות ידנית כלשהי.

#### להגשה (במסגרת מסמך ה-pdf):

- א. תיאור קצר (כחצי עמוד) של האלגוריתם שיכלול רשימה ממוספרת של הצעדים. זה לא צריך להיות ברמת פירוט של פסאודו-קוד, אלא באופן מתומצת כך שניתן להבין מה עשיתם.
- ב. תיאור קצר (חצי עמוד – עמוד) של בחירות מיוחדות שעשיתם בתכנון ובמימוש וההשפעה שלהן על התוצאות.
- ג. דיון קצר (כחצי עמוד) על איכות התוצאה (אפשר להתייחס לפאזלים מסוימים ספציפית), הסיבות לקשיים והצעה של כוונים לשיפור.

#### להגשה (בקובץ results.zip):

- א. תיקייה אחת לכל פאזל ובתוכה:
  - I. תמונה אחת של פתרון הפסיפס (כמו בדוגמא למעלה). שם הקובץ יהיה solution\_t1\_t2.jpeg כאשר t1 הוא מספר החלקים שהצלחתם להרכיב מתוך t2 החלקים שבפאזל הספציפי
  - II. תמונות של המיקומים היחסיים של החלקים שפתרתם (כמו בדוגמא למעלה). תמונה אחת לכל חלק עם השם piece\_i\_relative.jpeg עבור החלק ה-i.
  - III. תמונה אחת של הכיסוי שקיבלתם (לפי הדוגמא של "coverage count" למעלה)