

תרגיל בית 3

יצירת תמונה פנורמית

תאריך ההגשה: 30.12.2021

תאריך בדיקה פרונטלית: 2.01.2022 (טנטטיבית)



האם תהיתם פעם מה מתבצע מתחת לפני השטח כאשר מצלמים תמונה במצב פנורמה? טכניקות ראייה ממוחשבת מרובות מעורבות ביצירת פנורמה פשוטה. בעוד שכדי ליצור פנורמה טובה, משתמשים בהרבה טכניקות, העיקרון הבסיסי הוא למצוא keypoints בתמונות, למצוא התאמה של keypoints ולבצע "תפירה" של תמונות.

בתרגיל זה, נתרגל ליצור תמונה פנורמית משתי תמונות בלבד. בהינתן זוג תמונות כמו אלה שלמעלה, אנו "נתפור" אותן כדי ליצור סצנה פנורמית. חשוב לציין ששתי התמונות צריכות לשתף אזור משותף כלשהו. כמו-כן, כדי לאתגר את הבעיה, הפתרון שלנו יצטרך לעבוד גם אם לתמונות יש הבדלים באחד או יותר מההיבטים הבאים:

- Scaling
- זווית

בואו נבין תחילה את המושג תפירת תמונות. בעיקרון, אם אנחנו רוצים לצלם סצנה גדולה, אנחנו יכולים לצלם מספר תמונות של כל הסצנה ואז לשים את כל החלקים יחד לתמונה אחת גדולה. כך נקבל תצלום יפהפה אחד גדול! תמונות שנתפרו ממספר חלקים כאלו נקראות פנורמות.

בתרגיל זה, ניצור תמונה פנורמית משתי תמונות בלבד, שנשמך אותן על ידי left image ו- right image. שימו לב: סדר מאוד חשוב.

אלגוריתם תפירה שלנו יהיה מורכב ממספר צעדים.

1. חישוב נקודות עניין והמתארים שלהן (keypoints and their descriptors) עבור כל אחת מהתמונות.
2. חישוב מרחקים בין כל מתאר בתמונה אחת לכל מתאר בתמונה השנייה.
3. חישוב מטריצת הומוגרפיה (homography matrix).
4. ביצוע טרנספורמציה באמצעות מטריצת הומוגרפיה מצעד 3.
5. לבסוף, תפירת שני התמונות לתמונה פנורמית אחת גדולה.

נתאר כל אחד מהצעדים הללו בפרטים.

צעד 1 - חישוב נקודות עניין והמתארים שלהן עבור כל אחת מהתמונות

בשיעור למדנו על SIFT. בתרגיל זה תשתמשו באלטרנטיבה של SIFT – ORB. תוכלו למצוא הסבר קצר על ORB ומימוש ב-OpenCV בלינק הבא: https://docs.opencv.org/4.5.2/d1/d89/tutorial_py_orb.html. לא נדרש להבין את פרטי פרטים, רק את הרעיון הכללי.

נקרא את שתי התמונות מהדיסק ונמיר אותן לגונוני אפור (grayscale). חשוב שתמונת יהיו בגובה זהה. אם התמונות בגבהים שונים, יש לבצע resize לגובה אחיד. הרוחב יכול להיות שונה, אך הגובה חייב להיות זהה בשתייהן. חשוב: אם משתמשים ב-resize ומשנים את גובה של תמונה – יש לשמור על יחס גובה/רוחב. למשל, אם תמונה מקורית בגודל 100×200 ורוצים להגדיל את הגובה פי שניים, אז יש להגדיל גם את הרוחב פי שניים, כלומר התמונה המתקבלת תהיה בגודל 200×400 .

ניצור אובייקט לחילוץ נקודות עניין ORB והמתארים שלהן.

```
orb = cv2.ORB_create()
```

עכשיו, נקרא למתודה `orb.detectAndCompute(img1, None)` keypoints, descriptors =

עם כל אחת מהתמונות כדי לחשב נקודות עניין ORB והמתארים שלהן.

צעד 2- חישוב מרחקים בין כל מתאר (DESCRIPTOR) בתמונה אחת לכל מתאר בתמונה השנייה

כפי שתוכלו לראות, מצעד 1 מוחזר מספר רב של keypoints משתי התמונות. כעת, נשווה בין שתי קבוצות ה-descriptors ונבחר זוגות המראים דמיון רב.

ב-OpenCV, התאמת ה-descriptors מחייבת יצירת אובייקט Matcher.

```
matcher = cv2.BFMatcher(...)
```

ה- BFMatcher (BruteForce) עושה בדיוק את מה ששמו אומר. בהינתן שתי קבוצות של descriptors (מתמונת left ומתמונת right), כל descriptor מ- left מושווה מול כל descriptor מ- right. כברירת המחדל, BFMatcher מחשב את המרחק האוקלידי בין שני descriptors. עליכם לחקור איזו פונקציית מרחק מתאימה ביותר עבור מתארי ORB ולהשתמש בה.

עכשיו, נשתמש במתודה `match` למציאת ההתאמה בין שתי קבוצות מתארים.

```
matches = matcher.match(descriptors1, descriptors2)
```

מיינו את ה-matches לפי מרחק והשאירו רק 20% עם מרחק הנמוך ביותר.

בתום שלב זה, מומלץ לצייר את קווי ההתאמה בין שתי תמונות כדי לראות כיצד זה נראה כאשר מצאנו התאמה בתמונות.

התשמשו ב-

```
imMatches = cv2.drawMatches(im1, keypoints1, im2, keypoints2,
                             matches, None)
```



צעד 3 - חישוב מטריצת הומוגרפיה (HOMOGRAPHY MATRIX)

בצעד הקודם קיבלנו זוגות keypoints ומתארים שלהן המתאימים ביותר משתי התמונות. כעת, עלינו לקחת את ה-keypoints הללו ולמצוא את מטריצת הטרנספורמציה שבעזרתה נתפור את שתי התמונות על בסיס נקודות ההתאמה שלהן.

מטריצה כזו נקראת מטריצת הומוגרפיה. בקצרה, ההומוגרפיה היא מטריצה בגודל 3×3 שמייצגת טרנספורמציה דו-מימדית. טרנספורמציה זו ממפה נקודות ממישור אחד (תמונה) למישור אחר. בואו נראה איך אנחנו מבצעים זאת.

ניתן לחשב הומוגרפיה כשיש לנו לפחות 4 נקודות תואמות בשתי תמונות. ההתאמות שיצרנו בצעד הקודם לא תמיד מייצרות התאמות מדויקות ב-100%. לא נדיר ש-20-30% מההתאמות אינן נכונות. למרבה המזל, שיטת findHomography משתמשת בטכניקת הערכה חזקה הנקראת קונצנזוס מדגם אקראי (RANSAC) המייצרת את התוצאה הנכונה גם בנוכחות של התאמות גרועות.

```
H, status = cv2.findHomography(right_image_kp, left_image_kp, cv2.RANSAC)
```

כאשר left_image_kp, right_image_kp הם וקטורים של נקודות H-י היא מטריצת הומוגרפיה. שימו לב לסדר הפרמטרים.

צעד 4 - ביצוע טרנספורמציה באמצעות מטריצת הומוגרפיה

ברגע שיש לנו את ההומוגרפיה מצעד 3, עלינו להעביר את אחת התמונות למישור משותף באמצעות הטרנספורמציה. לשם כך אנו יכולים להשתמש בפונקציה warpPerspective().

```
res = cv2.warpPerspective(right_image, H, (width_panorama, height_panorama))
```

cv2.warpPerspective מקבלת כקלט התמונה עליה נפעיל טרנספורמציה, מטריצת H המתארת את הטרנספורמציה, וגודל תמונת הפלט. **שימו לב:** הפרמטרים width_panorama, height_panorama הם רוחב וגובה תמונת הפנורמה, כלומר גודל הפלט.

- הגדירו גובה תמונת הפנורמה, height_panorama, להיות גובה של התמונה השמאלית (להזכירכם, בצעד 1 העברנו את תמונות לגובה זהה).
- הגדירו רוחב תמונת הפנורמה, width_panorama, להיות סכום הרוחבים של שתי התמונות.



צעד 5 - תפירת שני התמונות לתמונה פנורמית אחת גדולה

השלב האחרון הוא לתפור את ששתי התמונות לתמונה אחת גדולה.

תהי"res תמונה שהתקבלה מצעד 4.

מה שנשאר הוא לאפס את החלק השמאלי של res ולשים בחלק זה את תמונת left image.

וזאת התוצאה:



שימו לב למספר דברים:

- בגלל הבדלי זווית יכולים להיות אזורים שחורים במסגרת התמונה. עליכם לזהות את האזורים השחורים ו"לחתוך" את התמונה, כך שנקבל את התוצאה הבאה:



- במידה ויש הבדלים חדים בזווית ותאורה, ייתכן ונראה "קו" באזור התפורה, כמו שניתן לראות בשתי התמונות הבאות. לא נטפל בתרגיל זה בבעיה זו. אבל אני מזמינה לחשוב איך ניתן לטפל בה.



קו באזור התפירה

הרצת התוכנית תתבצע משורת הפקודה בפורמט:

```
> Panorama.py path_left_img path_right_img path_output
```

כאשר Panorama.py הוא שם התוכנית, path_left_img, path_right_img הם מסלולים לתמונה שמאלית ותמונה ימנית, ו-path_output הוא מסלול לפלט.

אתם מוזמנים להריץ את הקוד על תמונות שלכם וליצור פנורמות.

הגשה:

יש להגיש קובץ zip ששמו מורכב מת"ז של המגיש/ים. קובץ ה-zip יכיל:

- קובץ/ים קוד עם התוכנית

- קובץ [readme.txt](#)

אופן הבדיקה:

הבדיקה תתבצע בצורה פרונטלית. מועדי הבדיקה ייקבעו בהמשך.

תהנו!!