

## Assignment 2:

# Homography & Panorama

Due Date: 25/12/18

### רקע

נתונות זוג תמונות (src.jpg, dst.jpg) וקובץ של נקודות תואמות (matches.mat). נרצה לבצע התמרה פרויקטיבית על תמונת המקור על מנת למזג אותה עם תמונת היעד, ולבנות מהן תמונת פנורמה.

כמוטיבציה לתוצאה הסופית, ניתן להתבונן באתר הבא:

<http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html>

במהלך התרגיל, נתחיל מבניית מערכת לחישוב הומוגרפיה (התמרה פרויקטיבית 2D) מתוך רשימת נקודות תואמות. בהמשך, נסיף למערכת יכולת התמודדות עם Outliers. לסיום, נשתמש במערכת זו לבניית פנורמה באופן (חצי) אוטומטי.

התרגיל ימומש ב python 3.6 ומעלה. מומלץ להשתמש ב-PyCharm Community (להורדה) או ב-IDE אחר לבחירתכם.

החבילות לשימוש בתרגיל: opencv, time, matplotlib, random, scipy, numpy, בשימוש בספריות נוספות יש לאשר מול עוזר ההוראה.

מספר פעולות הכנה:

- התחילו בטעינת התמונות src.jpg, dst.jpg.
- טענו את הקובץ matches\_perfect.mat. שימו לב כי התווספו שני משתנים: match\_p\_src ו-match\_p\_dst. משתנים אלה מכילים כל אחד 2 שורות ו-N עמודות, כאשר העמודות ה-i מייצגות קואורדינטות של נקודת התאמה i בתמונות ה-src וה-dst, בהתאמה.
- הציגו את הנקודות על גבי שתי התמונות, ובדקו שהן אכן מתאימות.
- כעת טענו את הקובץ matches.mat, המכיל בנוסף לנקודות ההתאמה גם מספר נקודות התאמה שגויות. הציגו כעת את הנקודות על גבי שתי התמונות, ושימו לב לנקודות השגויות.
- טיפ - לציור נקודות על תמונה, ניתן להשתמש בפונקציה matplotlib.pyplot.scatter.

**הערה** – את כלל הפונקציות אותן תידרשו לכתוב בתרגיל יש לאחד בקובץ אחד בשם **ex2\_functions.py**.

## חלק א': חישוב הומוגרפיה

1. בנו מערכת משוואות מהצורה  $A\underline{x} = \underline{b}$ , כפי שנלמד בכיתה, עבור התמרה פרויקטיבית. צרפו את הפיתוח לפתרון התרגיל. כיצד מקבלים ממערכת המשוואות את מטריצת ההמרה?
2. בנו פונקציה המשערכת מקדמי ההתמרה מהמקור (src) ליעד (dst), מתוך מערכת המשוואות שבסעיף 1. השתמשו ב-API הבא:

**def compute\_homography\_naive(mp\_src, mp\_dst):**

	<b>קלט:</b>
mp_src	– משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-src.
mp_dst	– משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-dst.
H	<b>פלט:</b>
	– מטריצת התמרה פרויקטיבית מ-src ל-dst.

3. טענו את הקובץ matches\_perfect.mat וחשבו את מקדמי ההתמרה בעזרת הפונקציה compute\_homography\_naive. הציגו את התוצאה.
4. ממשו פונקציית התמרה מהמקור ליעד באמצעות התמרת Forward Mapping, והציגו את תמונת המקור לאחר התמרה פרויקטיבית, בהתאם למקדמים שקיבלתם בסעיף 3.
5. מהן הבעיות ב-Forward Mapping וכיצד הן משתקפות בתמונה שקיבלתם?
6. כעת טענו את הקובץ matches.mat, וחזרו על סעיפים 3 ו-4. האם קיבלתם תוצאה שונה? הסבירו. ייתכן והתמונה תהיה גדולה מדי להצגה, ציינו זאת. (ניתן להקטין אותה ואז להציגה)

## חלק ב': התמודדות עם outliers

7. בנו פונקציה המחשבת את טיב המודל של התמרה פרויקטיבית. השתמשו ב-API הבא:

**def test\_homography(H, mp\_src, mp\_dst, max\_err):**

**קלט:**

- H – מטריצת התמרה פרויקטיבית מ-src ל-dst.
- mp\_src – משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-src.
- mp\_dst – משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-dst.
- max\_err – סקלאר המייצג מרחק מקסימאלי (בפיקסלים) בין נקודת src מותמרת לנקודת dst המתאימה לה, על מנת שתחשב כהתמרה וולידית (inlier).

**פלט:**

- fit\_percent – אחוז הנקודות (בין 0 ל-1) המותמרות בצורה וולידית.
- dist\_mse – שגיאה ריבועית ממוצעת של המרחקים בין נקודות src המותמרות בצורה וולידית לנקודות ה-dst המתאימות להן.

8. בנו פונקציה המחשבת מקדמי ההתמרה מהמקור ליעד, המתמודדת עם outliers באמצעות שימוש ב-RANSAC (השתמשו בפונקציות שבניתם בסעיפים קודמים). השתמשו ב-API הבא:

**def compute\_homography(mp\_src, mp\_dst, inliers\_percent, max\_err):**

**קלט:**

- mp\_src – משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-src.
- mp\_dst – משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-dst.
- inliers\_percent – האחוז הצפוי של נקודות התאמה נכונות מרשימת נקודות ההתאמה (בין 0 ל-1).
- max\_err – סקלאר המייצג מרחק מקסימאלי (בפיקסלים) בין נקודת src מותמרת לנקודת dst המתאימה לה, על מנת שתחשב כהתמרה וולידית.

**פלט:**

- H – מטריצת התמרה פרויקטיבית מ-src ל-dst.

9. נביח שישנן 30 נקודות התאמה, ונתון ש-80% מהן נכונות. מהו מספר ההגרלות שעלינו לערוך במקרה זה על מנת להבטיח הצלחה בשיעור של 90%? של 99%? כמה הגרלות יש להגריל על מנת לכסות את כל האפשרויות?

10. טענו את הקובץ matches.mat, וחשבו מקדמי ההתמרה בעזרת הפונק' compute\_homography. הציגו את מקדמי ההתמרה שהתקבלו, וכן תמונת המקור לאחר התמרה פרויקטיבית באמצעות Forward Mapping. השוו את התוצאות שקיבלתם לתוצאות בסעיפים 4 ו-6.

## חלק ג': בניית פנורמה

11. ממשו פונקציית התמרה מהמקור ליעד באמצעות התמרת Backward Mapping, העושה שימוש באינטרפולציה בי-ליניארית, והציגו את תמונת המקור לאחר התמרה פרויקטיבית בהתאם למקדמים שקיבלתם בסעיף 10. השוו לתמונה המתקבלת בסעיף 10.

12. בנו פונקציה המייצרת תמונת פנורמה משתי תמונות, ושתי רשימות של נקודות תואמות, המתמודדת עם outliers באמצעות RANSAC (השתמשו בפונקציות שבניתם בסעיפים קודמים). השתמשו ב-API הבא:

**def panorama(img\_src, img\_dst, mp\_src, mp\_dst, inliers\_percent, max\_err):**

**קלט:**

- img\_src – תמונת מקור הצפויה לעבור התמרה פרויקטיבית.
- img\_dst – תמונת היעד אליה מתמירים את תמונת המקור.
- mp\_src – משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-src.
- mp\_dst – משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-dst.
- inliers\_percent – האחוז הצפוי של נקודות התאמה **בכונות** מרשימת נקודות ההתאמה (בין 0 ל-1).
- max\_err – סקלאר המייצג מרחק מקסימאלי (בפיקסלים) בין נקודת src מותמרת לנקודת dst המתאימה לה, על מנת שתחשב כהתמרה וולידית.

**פלט:**

- img\_pan – תמונת פנורמה מאוחדת משתי תמונות הקלט.

**הדרכה:**

- השתמשו בהתמרת Forward Mapping על פינות תמונת המקור על מנת לייצר מלבן חוסם לתמונת התוצאה.
- השתמשו ב-Backward Mapping לביצוע ההתמרה.
- באזורי החפיפה בין התמונות, בחרו בערכים מתמונת היעד.

13. הפעילו את הפונקציה panorama על התמונות src.jpg ו-dst.jpg, והשתמשו בנקודות ההתאמה מהקובץ matches.mat. קבעו inliers 80% ושגיאה מקסימאלית של 25 פיקסלים. הציגו את תמונת הפנורמה שהתקבלה.

14. השתמשו בזוג תמונות שלכם לבניית פנורמה באמצעות פונקציית panorama. יש לקרוא לתמונות src\_test.jpg ו-dst\_test.jpg. שימו לב כי בשלב זה עדיף להשתמש בתמונות באותו גודל. הריצו את הקובץ create\_matching\_points.py על מנת לייצר את קובץ הנקודות matches\_test.mat עם 25 נקודות התאמה. יש לדאוג לכך שיהיו לפחות 10% נקודות התאמה שגויות ברשימה (outliers). הציגו את תמונות הקלט, יחד עם נקודות ההתאמה שסומנו, והציגו את תמונת הפנורמה שהתקבלה.

## הוראות הגשה:

- יש להגיש מסמך המכיל התייחסות לכלל הסעיפים בתרגיל, המציג את כל התוצאות ועונה על כל השאלות (אין צורך להציג קוד של פונקציות).
- יש לצרף את כל פונקציות ה-python שהוגדר להן API בתרגיל, וכן כל הפונקציות הנלוות אליהן שכתבתם (כולן צריכות להיות ב- **ex2\_functions.py**). בדיקת התרגיל תכיל הרצה אוטומטית של פונקציות אלו ובדיקתן.
- יש לצרף את זוג התמונות שלכם (`src_test.jpg`-`dst_test.jpg`) וכן את קובץ נקודות ההתאמה שלכם (`matches_test.mat`).
- יש לבדוק שאתם מצליחים להריץ את **test\_script.py** המצורף, ללא ביצוע שינויים בו. דבר זה יתאפשר רק אם כל הפונקציות אותן נדרשתם לכתוב בתרגיל נמצאות ב- `ex2_functions.py`.
- את הפתרון עם כל הקבצים הרלוונטיים יש להגיש בתיבת ההגשה במודל בתוך קובץ זיפ בשם הבא:**  
**assignment2\_ID1\_<id\_1>\_ID2\_<id\_2>**  
במידה וקובץ הזיפ עולה על 50MB יש לשלוח למייל : `mayamayberg@mail.tau.ac.il`  
את הנושא של המייל יש לנסח באופן הבא:  
`Assignment #2 ID1:<your_id_number> ID2:<your_id_number>`
- איחור בהגשה במועד יגרור הורדה בציון.

בהצלחה!