

# Assignment 1:

## Homography & Panorama

Due Date: 21/11/2014 @ 14:00

### 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 אחודה.

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.

מספר פעולות הכנה:

- התחילו בטעינת התמונות ל-MATLAB.
- טענו את הקובץ matches\_perfect.mat ב-MATLAB. 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 קוארדינטות 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.

## חלק א': חישוב הומוגרפיה

1. בנו מערכת משוואות מהצורה  $Ax = b$ , 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:

**H = compute\_homography\_naive(mp\_src, mp\_dst)**

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.

פלט:

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

3. טענו את הקובץ matches\_perfect.mat וחשבו את מקדמי ההתמרה בעזרת הפונק' 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.

## חלק ב': התמודדות עם outliers

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

**[fit\_percent, dist\_mse] = test\_homography(H, mp\_src, mp\_dst, max\_err)**

|             |                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------|
| <b>קלט:</b> |                                                                                                                 |
| 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     | – סקלר המייצג מרחק מקסימאלי (בפיקסלים) בין נקודת src מותמרת לנקודת dst המתאימה לה, על מנת שתחשב כהתמרה וולידית. |
| <b>פלט:</b> |                                                                                                                 |
| 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:

**H = compute\_homography(mp\_src, mp\_dst, inliers\_percent, max\_err)**

|                 |                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------|
| <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.                       |
| inliers_percent | – האחוז הצפוי של נקודות התאמה <u>נכונות</u> מרשימת נקודות ההתאמה (בין 0 ל-1).                                   |
| max_err         | – סקלר המייצג מרחק מקסימאלי (בפיקסלים) בין נקודת src מותמרת לנקודת dst המתאימה לה, על מנת שתחשב כהתמרה וולידית. |
| <b>פלט:</b>     |                                                                                                                 |
| H               | – מטריצת התמרה פרויקטיבית מ-src ל-dst.                                                                          |

9. נניח שישנן 30 נקודות התאמה, ונתון ש-80% מהן נכונות. מהו מספר הגרלות 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:

`img_pan = panorama(img_src, img_dst, mp_src, mp_dst, inliers_percent, max_err)`

|                              |                                                                                                                 |
|------------------------------|-----------------------------------------------------------------------------------------------------------------|
| קלט:                         |                                                                                                                 |
| <code>img_src</code>         | – תמונת מקור הצפויה לעבור התמרה פרויקטיבית.                                                                     |
| <code>img_dst</code>         | – תמונת היעד אליה מתמירים את תמונת המקור.                                                                       |
| <code>mp_src</code>          | – משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-src.                       |
| <code>mp_dst</code>          | – משתנה המכיל 2 שורות ו-N עמודות, בו העמודה ה-i מייצגת קואר' של נק' התאמה i בתמונת ה-dst.                       |
| <code>inliers_percent</code> | – האחוז הצפוי של נקודות התאמה <u>נכונות</u> מרשימת נקודות ההתאמה (בין 0 ל-1).                                   |
| <code>max_err</code>         | – סקלר המייצג מרחק מקסימאלי (בפיקסלים) בין נקודת src מותמרת לנקודת dst המתאימה לה, על מנת שתחשב כהתמרה וולידית. |
| פלט:                         |                                                                                                                 |
| <code>img_pan</code>         | – 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`. לקובץ הנקודות יש לקרוא `matches_test.mat`. יש לדאוג לכך שיהיו לפחות 10% נקודות התאמה שגויות ברשימה. הציגו את תמונות הקלט, יחד עם נקודות ההתאמה שסומנו, והציגו את תמונת הפנורמה שהתקבלה.

טיפים:

- לסימון נקודות, ניתן להיעזר בפונקציה `getpts`.
- לציור נקודות על תמונה, ניתן להשתמש בפונקציה `plot`.
- להוספת טקסט על תמונה (למשל מספור נקודות), ניתן להשתמש בפונקציה `text`.

## 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-MATLAB שהוגדר להן API בתרגיל, וכן כל הפונקציות הנלוות אליהן שכתבתם. בדיקת התרגיל תכיל הרצה אוטומטית של פונקציות אלו ובדיקתן.
- יש לצרף את זוג התמונות שלכם (src\_test.jpg ו-dst\_test.jpg) וכן את קובץ נקודות ההתאמה שלכם (matches\_test.mat).
- יש לבדוק שאתם מצליחים להריץ את test\_script.m המצורף, ללא ביצוע שינויים בו, וכאשר תיקיית העבודה של MATLAB היא התיקייה בה נמצא הקובץ.
- בכדי לא לאבד נקודות, מומלץ לבדוק שהקוד שכתבתם פועל גם על מחשב בלתי תלוי במחשב בו כתבתם אותו (למשל במעבדות באוניברסיטה), ולוודא שהרצה של כל אחת מהפונקציות מניבה תוצאה רצויה.
- **את הפתרון עם כל הקבצים הרלוונטיים יש להגיש למייל: [cv.apps.tau.2014@gmail.com](mailto:cv.apps.tau.2014@gmail.com) בלבד!**  
**את הנושא של המייל יש לנסח באופן הבא:**  
**Assignment #1 ID:<your\_id\_number>**
- איחור בהגשה במועד יגרור הורדה בציון.
- שימו לב! ייתכן וחלק מהתמונות שלכם יועלו לאתר הקורס. מי שמעוניין להימנע מכך, נא לציין זאת בעת הגשת התרגיל.

בהצלחה!