

# תרגיל 2

## הנחיות:

- 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.
- בכל שאלה, בנוסף להגשה של כל התוצרים וההסברים הנדרשים, יש להגיש את כל הקוד שאתם כתבתם (לא כולל קוד מספריות אחרות שנעזרתם בהן).
- תמונות הקלט לכל השאלות נמצאות בתיקיות Q1, Q2, Q3.
- בנוסף לקוד המימוש של כל שאלה, יש להגיש מסמך אחד, עדיף בפורמט pdf, עם כל התוצרים וההסברים הנלווים לכל שאלה, שבו יש לציין שם ותעודת זהות.
- את התרגיל יש להגיש במייל בכתובת [cv.checker.21@gmail.com](mailto:cv.checker.21@gmail.com) עם הנושא "תרגיל בית 2". 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 חמישי ה-3 ביוני בשעה 23:59. כל יום איחור יגרור הורדה של 10 נק'. בקשות מיוחדות לדחייה ניתן לשלוח מראש לאותו המייל.
- שימו לב שיש לעבוד עצמאית (בזוגות) ולממש בעצמכם את החלקים הנדרשים בקוד. תהיה תשומת לב לעניין בבדיקה של התרגיל.

## שאלות:

### 1. (35 נק') Epipolar Geometry

המטרה היא להתאמן בהפעלה ובבדיקת איכות של fundamental matrix משוערך. נממש ויזואליזציה של קוים אפיפולריים המתאימים לסט של נקודות מתאימות ונחשב שגיאה באמצעות מרחקים אלגבריים ואפיפולריים.

#### שליבים:

- א. לכל זוג תמונות, סמנו 8 (או יותר) זוגות של נקודות מתאימות. דאגו לכם שהן מפוזרות היטב במרחב התמונה וכי הן לא יושבות על מישור יחיד בסצנה.
- ב. חשבו את ה fundamental matrix בעזרת 2 שיטות estimation מתוך האפשרויות הבאות (ניתן להשתמש במימושים קיימים):

- 7-point algorithm
- 8-point algorithm
- normalized 8-point algorithm

ג. לכל אחת מהשיטות, הציגו את זוגות הנקודות ואת הישרים האפיפולריים המתאימים לכל נקודה בתמונה שמנגד. השתמשו בקוד משלכם להפעלת ה fundamental matrix (ע"י כפל מטריצה). ניתן להשתמש בקוד קיים לציור הקווים האפיפולריים שקיבלתם (במימוש שלכם יתכן ותצטרכו לחשב את קצוות הקווים שהם נקודות החיתוך עם מסגרת התמונה). ראו דוגמא מצורפת. ד. לכל אחת מהשיטות, חשבו את מדדי השגיאה הבאים (ע"י קוד שלכם), כממוצע על זוגות ההתאמות  $(x, x')$  שיצרתם:

- מרחק אלגבראי:  $x'^T Fx$
- מרחק אפיפולרי (סימטרי):  $d(x', Fx)^2 + d(x, F^T x')^2$ , כאשר אם נסמן ב  $l = (a, b, c)^T$  את הישר  $Fx$ , אז  $d(x', Fx) = x'^T Fx / \sqrt{a^2 + b^2}$  (שימו לב שאתם מבינים למה זהו המרחק האוקלידי בין כל נקודה והישר המתאים, שאמור להיות אפס בתנאים מושלמים).

### להגשה:

1. ויזואליזציות של סעיף ג. לכל אחת מהשיטות ולכל זוג תמונות.
2. טבלה של המרחקים השונים מסעיף ד. לכל אחת מהשיטות (ממוצע על פני זוגות התמונות).

## 2. (30 נק') Photometric Stereo

המטרה היא לחשב את מפת ה disparities המתאימה לתמונה השמאלית של rectified stereo pair בעזרת אלגוריתם פשוט מסוג scan-line.

### שליבים:

- א. לכל זוג שורות מתאימות, חשבו שורה אופטימלית של disparities, בנפרד עבור שיטות ה Sum-Square-Differences ו Normalized-Cross-Correlation עבור patches מגודל  $k \times k$  (עבור  $k = 3, 9, 15$ ).
- ב. השוו את מפות ה disparity עם מפות ה ground-truth בעזרת ארבע מדדי מרחק (שימו לב: יש לחלק ב-3 את ערכי ה ground truth disparity לפני שימוש):

1. AvgErr - mean of absolute differences in pixels
2. MedErr - median of absolute differences in pixels
3. Bad05 - percentage of disparities whose error is above 0.5
4. Bad4 - percentage of disparities whose error is above 4

### להגשה:

1. לכל אחת מ 6 הקומבינציות של  $\{SSD, NCC\} \times \{k=3, 9, 15\}$  הציגו את מפת ה disparities לכל זוג תמונות, עם 4 סוגי מדדי ה error בכותרת של כל אחת מהתמונות.
2. דונו בקצרה בשילובים המועדפים עבור זוגות התמונות מבחינת איכות וזמן ריצה.

### 3. (35 נק') New View Synthesis

המטרה היא ליצור זוויות צפיה חדשות של סצנה, בהינתן תמונה ומפת עומק. ניתן לעשות זאת ע"י שינוי ה pose של המצלמה וסינתוז של התמונה שהייתה נרכשת מאותה זווית צילום חדשה.

#### שליבים:

א. חשבו reprojection של כל קואורדינטות התמונה (פיקסלים) אל המרחב התלת-ממדי. ניתן לעשות זאת בשימוש במטריצה  $K$  של ה intrinsics (ההופכית) ובמפת העומקים  $D$ . הנקודות בתלת-ממד יהיה במערכת הקואורדינטות של המצלמה, כלומר שאנחנו משתמשים במטריצת המצלמה  $P = K[R|T]$  כאשר  $R = I$ ,  $T = 0$ . (שימו לב: ישנן בתיקייה פונקציות פייתון ומטלב לקריאה של מטריצת מצלמה cam. ומפת עומקים dpt).

ב. הטילו בחזרה את הנקודות בתלת-מימד אל מישור המצלמה, במיקום המקורי שלה. כעת, סנתזו את התמונה המתקבלת ע"י העתקה של ערכי הפיקסלים ב RGB מהתמונה המקורית. התוצאה אמורה להיות זהה לתמונה המקורית - זוהי בסה"כ בדיקת נכונות של תהליך ה 2D-3D-2D.

ג. יצרו סדרת תמונות מזוויות חדשות על ידי שינוי ה pose של המצלמה. ניתן לעשות זאת ע"י מניפולציה של  $T$  ו  $R$  במטריצת ה extrinsics. ראו דוגמא לרפרנס בתיקייה המתאימה (אפשר לעבור על הרצף במהירות על ידי viewer, כמו irfanview, כך שיראה כמו וידאו). ביתר פירוט, אם נסמן ב  $M_0 = [R | T]$  את (המטריצה) של ה pose המקורי, סנתזו רצף תמונות jpeg (ניתן לשמור אותן בחצי או רבע רזולוציה) על ידי:

- סיבוב המצלמה הלוך וחזור סביב ציר ה-x
- סיבוב המצלמה הלוך וחזור סביב ציר ה-y
- סיבוב המצלמה הלוך וחזור סביב ציר ה-z
- הזזת המצלמה הלוך וחזור לאורך ציר ה-x
- הזזת המצלמה הלוך וחזור לאורך ציר ה-y
- הזזת המצלמה הלוך וחזור לאורך ציר ה-z

#### להגשה:

1. תמונות ה jpg המסונתזות לכל אחת משתי הסצנות (בחרו 2 מתוך ה 3). וודאו שאתם שומרים את רצפי התמונות בתת-תיקיות נפרדות פר סצנה.
  2. שימו לב להבדלים בין החלק של הסיבוב לזה של ההזזה.  
א. איזה משניהם נותן לצופה מידע על המבנה התלת-ממדי של הסצנה?  
ב. הסבירו מדוע.
  3. שימו לב ל"חורים" (אזורים מושחרים) בתוצאות. ישנם שני סוגים של חורים: (I) קוים דקים (כמו שניתן לראות למשל בפריימים 32 או 52 בדוגמא); (II) חורים משמעותיים יותר (כמו בפריימים 144 או 228).
- א. הסבירו את המקור של כל אחד משני סוגי החורים האלו
- ב. איזה משני הסוגים הוא אינהרנטי (כלומר נובע מחוסר אינפורמציה) ואיזה ניתן לפתור? (לגבי הפתרון - מספיק לתת הצעה לכך, מבלי לממש).