

ראייה ממוחשבת – תשפ"ב

תרגיל בית 2

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.
- תמונות הקלט ודאטה נוסף לכל השאלות נמצאות בתיקיות Q1, Q2.
- יש להגיש את כל הקוד שאתם כותבתם (לא כולל קוד מספריות אחרות שנעזרתם בהן) בקובץ יחיד בשם code.zip, שבתוכו חלוקה של תיקיות Q1, Q2.
- בנוסף לקוד המימוש של כל שאלה, 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.
- **נוהל חדש:** התרגיל להגשה עד ליום חמישי ה-26 במאי בשעה 23:59. כל יום איחור יגרור הורדה של 3 נק', עד לתאריך אחרון של יום שלישי ה-31 במאי. הפעם לא תינתן דחייה כלשהי (פרט למקרים מאושרים ספציפיים כגון מילואים).

שאלות:

1. (50 נק') Epipolar Geometry

לכל זוג תמונות בתיקיה Q1 בצעו את החישובים והויזואליזציות הבאות:

1. סמנו באופן ידני שני סטים (זרים) של זוגות של נקודות מתאימות (matches): סט אחד (S_1) ישמש לחישוב fundamental matrix והסט השני (S_2) ישמש כקבוצת ביקורת כדי לחשב את הדיוק של המטריצה.
 - a. בכל סט יש לכלול בדיוק 10 זוגות - אבל כאלה שניתן לסמן בדיוק טוב (פינות, למשל).
 - b. כדאי שהנקודות יהיו כמה שיותר מפוזרות במרחב התמונה, כך שיהיה דיוק בכל האזורים.
 - c. כדאי שהנקודות ישבו על מישורים שונים בעולם התלת-ממדי (כי אחרת מקבלים פתרון שלא מנצל את כל דרגות החופש של המודל)
2. הזינו את הזוגות של S_1 ל-solver של fundamental matrix כרצונכם, כזה שמבצע פתרון least-squares (כמו ה 8pt-algorithm) על כל הזוגות בבת אחת, ללא שימוש ב-RANSAC וקבלו fundamental matrix שנקרא לו F .
3. בצעו את הפעולות הבאות בנפרד עבור סט החישוב (S_1) ועבור סט הביקורת (S_2):
 - נסמן את קבוצת ההתאמות (הזוגות) בקבוצה על ידי $\{x_i, x_i'\}_{i=1}^{10}$ כאשר x_i נקודה (דו-ממדית) בתמונה השמאלית ו- x_i' הנקודה המתאימה בתמונה הימנית.
 - נסמן ב- l_i את הישר האפיפולרי בתמונה שמאל שמתאים לנקודה x_i' (מתמונה ימין)
 - נסמן ב- l_i' את הישר האפיפולרי בתמונה ימין שמתאים לנקודה x_i (מתמונה שמאל)
 - חשבו בעזרת המטריצה F את הישרים האפיפולריים $\{l_i\}_{i=1}^{10}$ ו- $\{l_i'\}_{i=1}^{10}$
 - הציגו את זוג התמונות זה לצד זה. ציירו על גביהן את 10 זוגות הנקודות בשימוש ב-10 צבעים שונים. ציירו את הישרים האפיפולריים $\{l_i\}_{i=1}^{10}$ ו- $\{l_i'\}_{i=1}^{10}$ בצבעים המתאימים. ראו את התמונה [vis_example.jpg](#) לדוגמא (שימו לב שזאת רק דוגמא לא אמיתית שצוירה ידנית – יש בה רק 5 זוגות והכל לחלוטין מומצא...). אתם צריכים לצייר בצורה ברורה ומדויקת.
 - חשבו את הדיוק הממוצע של המטריצה F על הסט, לפי ממוצע ה- [Symmetric Epipolar Distance \(SED\)](#), על פני 10 הזוגות. המדד הזה מחשב שגיאה באמצעות סכום ריבועי המרחקים בין הנקודות אל הקווים המתאימים (בשני הכוונים, כלומר בתמונה ימין ושמאל). כתבו את תוצאת המדד בכותרת של התמונה שאתם מציגים.
4. הסבירו את הבדלי הדיוק (SED וויזואלית), אם קיימים, בין המקרה של סט החישוב וסט הביקורת.

2. (50 נק') Metric Reconstruction

החישובים בשאלה זו יבוצעו על הדאטה שבתיקייה Q_2 :

- [house_1.png](#) ו- [house_2.png](#) – זוג תמונות הקלט
- [cameraMatrix1.txt](#) ו- [cameraMatrix2.txt](#) : מטריצות המצלמה המתאימות
- [matchedPoints1.txt](#) ו- [matchedPoints2.txt](#) : התאמות בין שתי התמונות
- [matches_connected.jpg](#) – תוצאת ויזואליזציה לדוגמא (בכוונה ברזולוציה נמוכה)
- [matches_xy_projected.jpg](#) – תוצאת ויזואליזציה לדוגמא (בכוונה ברזולוציה נמוכה)
- [reconstruction.gif](#) – תוצאת שחזור לדוגמא (תחת רוטציית 3D **אקראית** על הנקודות שאני בחרתי)

הנחיות:

- נתונות שתי תמונות של בית שצולמו משתי זוויות שונות
 - נתונות מטריצות המצלמה של שתי המצלמות
 - נתונה סדרה של כ-22 התאמות בין שתי התמונות
 - **שלב א':** ויזואליזציה של סדרת ההתאמות
1. לקרוא את הדאטה מהדיסק
 2. לצייר בעצמכם (בעזרת פונקציות ציור קיימות) את ההתאמות על זוג התמונות, ברזולוציה טובה, בהתאם לדוגמא ב [matches_connected.jpg](#)
- **שלב ב':** חישוב ויזואליזציה של השחזור המטרי של הנקודות
1. לבצע טריאנגולציה כדי לקבל את קואורדינטות ה-3D של 22 הנקודות, כמו שלמדנו בכיתה (בעזרת DLT מבוסס SVD). אפשר להשתמש בפונקציה מוכנה (כדוגמת [DLT כאן](#)), או [triangulate כאן](#).
 2. למרכז את ענן הנקודות על ידי החסרת הממוצע מכל הנקודות.
 3. לצייר את הנקודות באופן מקושר על ידי הטלה שלהם למישור ה-xy (כלומר פשוט להתעלם מקואורדינטת ה-z), בהתאם לדוגמא ב [matches_xy_projected.jpg](#)
 4. להפעיל רוטציית תלת-ממדית אקראית אחת על ענן הנקודות.
 5. לחזור על הציור של שלב 3 בשתי לולאות. הראשונה היא סדרה של 36 רוטציות תלת-ממדיות סביב ציר ה-x בזוויות של 0 עד $360 (2\pi)$ בקפיצות שוות של $10 (2\pi/36)$ שמבצעים על הנקודות לפני ההטלה והשמירה של התמונה. לאחר מכן סדרה זהה על ציר ה-y. בסה"כ נקבל 72 (או אולי 74) תמונות שמהן נבנה [animated gif](#), כמו בדוגמא [reconstruction.gif](#), רק שימו לב שלכל אחד מכם זה יצא אחרת (ושונה משלי) בגלל הרוטציה האקראית. את ה-gif אפשר ליצור ישירות מהקוד, אבל אולי יותר נוח לשמור תמונות jpg בתיקייה ואז להשתמש בקוד ייעודי (או באפליקציה כמו [ezgif.com](#)).