

ראייה ממוחשבת – תשפ"ב

תרגיל בית 1

הנחיות:

- 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.
 - יש להגיש את כל הקוד שאתם כתבתם (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 במייל בכתובת --TBD--@--TBD-- עם הנושא "תרגיל בית 1". 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 רביעי ה-13 באפריל בשעה 23:59. כל יום איחור יגרור הורדה של 10 נק'.
- בקשות מיוחדות לדחייה ניתן לשלוח מראש לאותו המייל.

שאלות:

1. (50 נק') Hough Transform on ellipses

המטרה היא למצוא אליפסות בתמונות טבעיות בעזרת וריאנט של Hough Transform (HT). תצטרכו לממש detector ולהפעיל אותו על התמונות בתיקייה ellipses, כאשר התוצאה תצויר על גבי התמונות. כל אליפסה שנמצאת צריכה להיות מצוירת (באופן אוטומטי ע"י הקוד). בכל תמונה תנסו לגלות את האליפסות בכל מיקום, גודל ואוריאנטציה אפשריים, גם כאלה שנראים באופן חלקי (הסתרה או חריגה מגבולות התמונה).

שלבים מוצעים:

- א. היזכרו במידול המתימטי הבסיסי של האליפסה בעזרת הערך <https://en.wikipedia.org/wiki/Ellipse>
- ב. חשבו edge map (ניתן להשתמש במימושים קיימים כלשהם, למשל של Canny detector) - אפשר לכוון את הפרמטרים של המימוש (אפילו פר תמונה) לשיפור התוצאה. ניתן לעשות post-processing כלשהו לניקוי (אבל לא באופן ידני).
- ג. חשבו את כיווני ה-edges בעזרת חישוב גרדיאנטים.

- ד. ממשו את ה-ellipse detector על ידי HT לפי הפרטים במאמר:
<http://www.bmva.org/bmvc/1988/avc-88-041.pdf> בהתאם לשלבים הבאים:
- מציאת מרכז האליפסה (בעזרת HT שמוסבר בפרק "2.1 Stage 1: Center Finding")
 - מציאת שאר פרמטרי האליפסה (בעזרת HT תלת-ממדי והיררכי שמוסבר בפרק "2.2 Stage 2: Determination of remaining parameters"). ניתן להתעלם ממשוואה (4) ומהפסקאות שמיד מעליה ומתחתיה. כלומר הפתרון שמתואר מתבסס על משוואה (3) וההסבר שבחלק השני של הפרק.
 - ה. הפעילו את האלגוריתם בשלמותו לגילוי וסימון האליפסות בתמונת קלט.

הערות:

- רצוי להעביר את התמונות ל-grayscale וניתן לשנות את הגודל שלהן.
- ניתן לשחק עם thresholds ופרמטרים לכל אורך התהליך. אפשר גם לבחור פרמטרים שונים עבור תמונות קלט שונות.
- ניתן להפעיל non-maximal suppression לשיפור התוצאה בשלב המתאים.
- הפעילו שיקול דעת בצורך לטפל במספר של אליפסות. האם לחשב הכל מחדש ובאיזו רמה עבור כל אליפסה. מה נותן תוצאות טובות יותר וביצועי זמן ריצה טובים.
- ניתן לצייר את האליפסה על ידי פונקציות קיימות שמקבלות את פרמטרי האליפסה. יש להשתמש בעובי קו דק בצבע בולט על גבי גרסת ה-grayscale של התמונה. חשוב: כל הציור צריך להיעשות באופן אוטומטי מתוך הקוד, בלי התערבות ידנית.
- שימו לב שחלק מהקלטים קשים במיוחד. לא ניתן להגיע לתוצאות מושלמות, אלא רק לנסות לדייק במידת האפשר.

להגשה:

1. תיאור קצר (פסקה) של בחירות מיוחדות שעשיתם במימוש וההשפעה שלהן. רצוי להתייחס לאופן שבחרתם לבצע את החישוב על כל האליפסות בתמונה (עם או בלי מקבול כלשהו).
2. עבור 10 מתוך 12 תמונות הקלט (לבחירתכם):
 - א. מפת ה-edges
 - ב. תמונת הקלט ועליה מצוירות כל אחת האליפסות
 - ג. דיון קצר (פסקה קצרה) על איכות התוצאה, הסיבות לכשלונות (false detections inaccurate) (detections, mis-detections) והתייחסות לאיך היה אפשר להתגבר עליהם.

2. (50 נק') Multi-View Denoising

עובדה ידועה היא שהחלשת רעש גאוסיאני אפשרית על ידי מיצוע של דגימות בלתי תלויות. לשם המחשה, נניח שאנחנו ממקמים מצלמה על חצובה ומצלמים n תמונות רועשות של אותה סצינה בתנאים של תאורה חלשה. ניתן למדל את התמונה ה- i על ידי $I_i = I + X_i$, כאשר I התמונה הנקייה ואילו X_i תמונה (מטריצה) מאותו גודל שבה כל כניסה נבחרה באקראי מתוך התפלגות נורמלית (גאוסית) עם תוחלת 0 וסטיית תקן (רמת רעש) σ . הממוצע של n התמונות ייתן תמונה חדשה $\bar{I} = I + \bar{X}$ עם רמת רעש של σ/\sqrt{n} (כלומר כל כניסה ב- \bar{X} מתפלגת נורמלי עם תוחלת 0 וסטיית תקן σ/\sqrt{n}).

המטרה היא, בעזרת העיקרון הזה, לנקות תמונה רועשת בעזרת תמונות (רועשות) אחרות שחופפות בשטחן. לשם כך, נצטרך לחשב את ה-warp של כל אחת מהתמונות אל תמונת המטרה, כלומר לחשב alignment ביניהן.

חלק 1: הפעילו את ה-warps הנדרשים על מנת להעתיק את כל אחת מהתמונות אל תמונת ה-target

שלבים מוצעים:

- א. מצאו מימוש קיים של SIFT והפעילו אותו על כל אחד מזוגות התמונות. תוצאת ה-detector, לכל נקודת עניין (IP), הוא וקטור $[x, y, r, t]$, כאשר x ו- y הם קואורדינטות מיקום ו- r ו- t הם ה-scale וה-orientation, ותוצאת ה-descriptor הוא וקטור 128 ממדי.
- ב. בהינתן שיש N ו- M נקודות עניין בזוג התמונות בהתאמה, חשבו את מטריצת המרחקים מגודל $N \times M$ בין כל זוגות וקטורי ה-descriptors.
- ג. חלצו את ההתאמות (matches) שעברו את מבחן ה-ratio-test (עם סף של 0.8)
- ד. מצאו מימוש קיים של solver מינימלי - כזה שמשתמש ב-4 matches להתאמת הומוגרפיה.
- ה. מצאו מימוש קיים של פונקציית warping, אשר יכולה להפעיל הומוגרפיה (רצוי לבחור אופציה מתקדמת של אינטרפולציה, כגון bilinear או bi-cubic).
- ו. ממשו לולאת RANSAC פשוטה עם solver ההומוגרפיה. כדאי לנסות לתת ל-solver בשלב שני את כל נקודות התמיכה כדי לחשב טרנספורמציה מדויקת יותר.
- ז. לכל זוג תמונות, העתיקו באמצעות warping את תמונת ה-source לתמונת ה-target, על בסיס הטרנספורמציה שהותאמה.

חלק 2: חשבו את תמונת הממוצעים הנקייה במרחב של תמונת ה-target.

שלבים מוצעים:

- א. חשבו את תמונת הממוצעים של תמונת ה-target עם התמונות הממופות (במערכת הקואורדינטות של תמונת ה-target. שימו לב שבכל מיקום (פיקסל) יכול להיות מספר שונה של ערכים שתורמים לממוצע, כי תמונה ממופית לא בהכרח מכסה את כל תמונת ה-target (ואם משתמשים באינטרפולציה מתקדמת התרומות הן מספרים לא שלמים). דרך אפשרית להתמודד עם התופעה הזאת היא למפות (עבור כל תמונת source) תמונה שכולה 1-ים אל מרחב תמונת ה-target ולהחזיק מטריצה שסופרת את מספר התרומות לכל פיקסל – מה שיאפשר חישוב ממוצע על ידי חלוקת סכום התרומות לפיקסל במספר התרומות לפיקסל.

להגשה:

1. לכל סדרת תמונות שבתיקייה denoising_sets, הציגו את התוצאה – תמונת ה-target המורעשת לצד תמונת ה-target ה'נקייה'.
2. דונו בקצרה באיכות התוצאה. למשל: איך היא הושפעה מאיכות ההענקות? ומכמות התרומות בכל איזור?