

עיבוד תמונה – תרגיל 4

אוריין חסידים ~ oryan.hassidim ~ 319131579

1. מבוא

בחורגיל זה נדרש לבצע **Image Alignment** – קיבלנו שתי תמונות מקבילות, שבאחת מהן הורדה הרזולוציה ובאחת מהן בוצע חיתוך ו- **warping**, ואנו נדרשים לישר את החתוכה אל התמונה עם הרזולוציה הנמוכה.

נשתמש בכלים אוחם לממדנו בכיתה, וספציפית בכיוון של **MOPS**: **חילוץ פיצרים עיי** זיהוי פינות באלאגוריתם **Harris**, חילוץ דסקריפטורים כחלון מרזולוציה נמוכה שמיושר לפי הגרדיאנט, אלג' **RANSAC** למציאת התאמת של פרמטרים לדגימות, ויישור תמונה. נשתדר עד כמה שאפשר למש את האלאגוריתמים המעניינים בעצמו ולא להשתמש במוכנים של **OpenCV**.

2. אלגוריתם

a. חילוץ פיצרים – זיהוי קצוות ע"פ אלגי Harris // מקבל תמונה, מוחזר ורשימת קצוזות

a. המרה לגונני אפור;

b. עبور גDALי חלון שונים (משתנה לפי אופי התמונה):

c. חישוב תగובה הפינה של הפיקסלים עבור חלון בגודל זה ע"פ הנוסחה של ממדנו, ונורמל לגודל החלון:

$$R(x, y) = \frac{1}{|W|} \cdot \sum_{(dx, dy) \in W} [I(x, y) - I(x + dx, y + dy)]^2$$

d. עبور כל פיקסל מציאה של התגובה המקסימלית;

e. בחירות כל התשובות מעל לסף מסוימים (סף יחסיב ביחס לתגובה המקסימלית על התמונה);

f. סינון הפיקסלים עבורם מתקבל מקסימום לוקאלי בתגובה הפינה;

g. מציאת האינדקסים של הפינות שנשאו.

ב'. חישוב דסקריפטורים – מציאת חלונות מאפיינים // מקבל תמונה ורשימת נקודות (קצוזות), ומוחזר רשימה של חלונות מאפיינים עבור כל פינה:

a. חישוב הגרדיאנט בפיקסל ע"פ גווני האפור של התמונה;

b. חישוב הזווית של הגרדיאנט;

c. לקיחת חלון בגודל 44×44 מסביב לפינה זו;

d. סיבוב החלון כדי לישר את הגרדיאנט כלפי מעלה;

e. לקיחת חלון בגודל 32×32 ;

f. הורדת שתי רמות בפרימידה;

g. נורמל ע"פ ממוצע וסטיית תקן.

ג'. מציאת התאמות בין דסקריפטורים // מקבל שתי רשימות חלונות מאפיינים, מוחזר ורשימה של זוגות אינדקסים מתאימים

עבור כל חלון מתוך אחת:

a. מציאת שני חלונות הכידומים (במרחק אוקלידי) בתמונה השנייה;

b. אם היחס בין המרחקים הוא מתחת לסת' שבחרנו – הכרזה על הזוג כמתאים

c. סינון ומיון מערכי הפינות לפי התאמות.

ד'. RANSAC // מקבל שתי רשימות קצוזות מסודרות לפי זוגות מתאימים, מוחזר מטריצה יישור מיטבית (רנדומית)

עבור מספר איטרציות:

a. הROLה של 4 התאמות רנדומליות;

b. חישוב מטריצה ההומוגרפיה לפי 4 התאמות המוגROLות;

c. בדיקה כמה מההתאמות האחרות, המרחק בין הציפייה למיושרת קטן מסף כלשהו.

הՁזה המטריצה עבורה היו התאמות רבות ביותר עם מרחק קטן מהסתף.

ה'. חישוב מטריצת הומוגרפית מ-4 נקודות // מקבל 4 זוגות נקודות, מוחזר מטריצת יישור
אנחנו מוחשים מטריצה מהצורה:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x'_1x_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x'_2x_2 & -x'_2y_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x'_3x_3 & -x'_3y_3 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x'_4x_4 & -x'_4y_4 \end{bmatrix}^{-1} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{bmatrix}$$

שתקיים עבור כל 4 הנקודות. אחרי כמה חישובים אנו מקבלים את המערכת הנ"ל ופתראים אותה.

ו'. **ישור פרספקטיבי של תמונה ע"פ מטריצה // מקבל תמונה ומטריצת יישור, מוחזר תמונה מיושרת**
עבור כל פיקסל בתמונה המטריצה:

- a. מחשבים את המקור ע"פ המטריצה ההפכית;
- b. מחשבים את ערכול הצלבים לפי הפיקסלים ביןיהם נפלנו.

ז'. **הדבקת שתי התמונות // מקבל שתי תמונות ומטריצת יישור, מוחזר תמונה ממוגנת**

- a. טעינת עוזץ השקיפות של התמונה עם הרזולוציה הגבוהה;
- b. ישור שלו ע"פ מטריצת היישור שמצאו;

c. קביעת המסכה כמעבר סף של עוזץ השקיפות המושר;

d. סכימה ביןארית ע"פ המסכה הבינארית של התמונה עם הרזולוציה הגבוהה מיושרת ע"פ היישור
שמצאו עם התמונה בעלת הרזולוציה הנמוכה.

ח'. **מציאת התאמה בין שתי תמונות // מקבל שתי תמונות, מוחזר תמונה ממוגנת בהתאם לקסימלית**
(rndom)

- i. עבור כל תמונה:
 - .i. טשטוש גאוסיאני;
 - .ii. מציאות פינות;
 - .iii. חלונות מאפיינים לכל פינה;
- ii. מציאת התאמות בין חלונות משנה תמונות;
- iii. RANSAC למציאת מטריצת היישור הטובה ביותר;
- iv. הדבקה של התמונה המיושרת לתמונה עם הרזולוציה הנמוכה.

3. פרטי מימוש

בריגל, השתמשתי ב-NumPy וב-OpenCV, משומם שהם מאד נוחים, מאד פופולריים, ועם דוקומנטציה הנדרת.

א'. **חילוץ פיצרים – זיהוי קצוות ע"פ אלג' Harris**
בחorthy להשתמש בקירוב של הא里斯 ע"י הפונקציה שתוארה לעיל באלגוריתם ולא ע"י הע"ע של מטריצת הנגורות החלקיות מרמה שנייה, עקב העובדה הקירוב קל בהרבה לחישוב ועשה עבודה מעולה על התמונות שניתנו.

لتמונה הראשונה גDAL החולנות היו [5,11,21,41], לתמונה השנייה [5,11], סף לתמונה הראשונה – 0.1, ולתמונה השנייה – 0.3. מקסימום לוקאלי ע"פ חלון בגודל 3×3 (רק מי שיש להם פאה או קדקך מסווטף).

מומש עצמאית.

אחרי המון סבבים עם תוצאות מוזרות למדי, גיליתי שהפתרונות הוא כמו בתרגילים הקודמים – המורה לא-
. int32 😊

ב'. **חישוב דסקריפטורים – מציאת חלונות מאפיינים**
גודל 44 לחלון לפני הסיבוב כדי שבכל מקרה אחרי הסיבוב יהיה חלון 32 מהקטע. $44 \leq 32 \cdot \sqrt{2}$.

הורדת רמה בפירמידה מספרייה OpenCV. כל השאר מומש ידנית.

ג'. מציאת התאמות בין דסקריפטורים לתמונה הראשונה סף יחס נמוך יחסית (ברן יותר) – 0.6, ולתמונה השנייה סף יחס גובה – 0.9. בדיעבד מסתבר שבטעות לא שמשי כלל סף ראשוני ושמתי סף יחסי בלבד, מה שהביא אליו אלו התאמות תמהות המשו, אבל ה-RANSAC פיצה על זה בכל מקרה.

מומש ידנית.

ד. RANSAC

לא כל פעם יהיה מבסוט מההוצאה אז לפעמים הרצוי שוב... או יש תמונות שזכו לפחות או יותר 10000 הרצות. סף המרחק המקסימלי המותר היה 5 לשתי התמונות. מומש ידנית.

ה'. חישוב מטריצת הומוגרפיה מ-4 נקודות השתמשתי בפונקציה המוכנה של OpenCV. לשם לב שהסטנדרט שלהם לגבי x ו-y הוא הפוך אז הינו צריכים להפוך לפני העבודה אותה.

ו. ישור פרספקטיבי של תמונה ע"פ מטריצה מאחר וכבר הינו בתענוג מעין זה בתרגיל 6 של קורס אינטרו (הפתרון שלי ב-[github](#)), השתמשתי בפונקציה המוכנה של OpenCV.

ז'. הדבקת שתי התמונות בהרתי לעשות את המסכה כסף של ערוץ השקיפות כדי להימנע מהקו השחור בקריאת הקובץ כ-RGB ורגיל. ערבול פירמידה לא עוזר כאן בגלל התדר הגובה שלו. ערבול פירמידה טוב באזוריים דומים. הסף הוא בערך 240 (מתוך 255). מומש ידנית.

ח'. מציאת התאמה בין שתי תמונות מומש ידנית באמצעות כל הנ"ל.

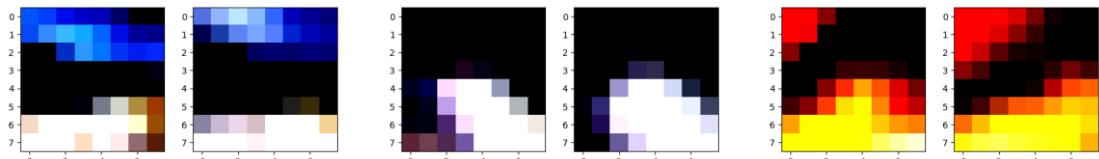
4. תוצאות ויזואליות

א. חילוץ פיצרים – זיהוי קצוטה ע"פ אלגי Harris



במבט ראשון נראה שאלו המונח "פינוט" שלוו דואק נכונות. מהתבוננות בפאצ'ים (להלן) נראה שם אכן מספיק מעניינות ומצדיקות יחס, ופושט ציריך רזולוציה גבוהה מספק כדי לשים לב לפיניטיותם. כמו כן, מctrur הזיהוי של האובייקט בתמונה עם הרזולוציה הגבוהה, אבל במקרה לנוכח לסנן אותן ידנית, ניתן למציאת התאמות לטפל בה.

ג'. מציאת התאמות בין דסקריפטורים דוגמאות לזוגות מתאימים:

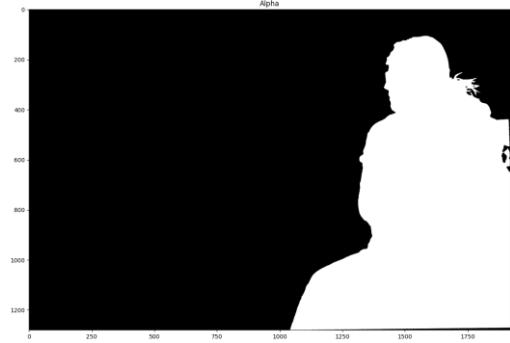


(כל הצבעים מנורמלים, אז החלוניות לא מופיעים בצדדים הילו במקור)



הימנית – כל התאמות, בשמאלית – התאמות שקצורות מ-400, שהוא מה שהגוני פה (لتצוגה בלבד). בעצם רוב הקווים המאונכים הם בגודל אלו שאנו ממחפשים.

ד. הדבקת שתי תמונות



(אין פה ממשו מעניין או מרגש מדי, זו המסכה ע"פ ערזן השקיפות המיושר)

ה/. מציאת התאמה בין שתי תמונות
להלן תוצאות סופיות:





5. מסקנות

בעבודה זו מימשנו אלגוריתמים מסוימים להתקאת ויזור תמונות (כן אני יודע שזה נשמע ניסוח של בינה מלאכותית, מבטיח שמש כתבתי את זה בעצמי, זו באמת התהוושה שלי).

אחד הדברים המעניינים הוא שאע"פ שבדייעד גיליתי שהו המון אי דוקים במימוש שלג, והמון מקומות שהיו יכולים לשפר מאוד את זמן הריצה ואת איכות ההתאמות, בעצם אלגוריתם ה-RANSAC חיפח על כל אלו והצליח להביא לחוצאות מרשים אע"פ שהקלט אליו לא היה מיטבי.

גם כאן, כרכי בחריגלים הקודמים, היו כל מיני טעויות טכניות (טיפוסים ב-NumPy, קונבנציית y,x ב-OpenCV) שאתגרו אותנו, אבל בסוף הצלחנו להתגבר.

תודה רבה !