

## עיבוד תמונה – תרגיל 4

אוריין חסידים ~ oryan.hassidim ~ 319131579

### 1. מבוא

בחറיגל זה נזכיר 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 ע"י זיהוי פינות/aggregation, Harris, חילוץ דסקריפטורים כחלון מרזולוציה נמוכה שמיושר לפי הגרדיאנט, אלג' RANSAC למציאת התאמת של פרמטרים לדגימות, וישור תמונה. נשתדר עד כמה שאפשר למש את האלגוריתמים המופיעים בעצמו ולא להשתמש במוכנים של OpenCV.

### 2. אלגוריתם

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

a. המרה לגונני אפור;

b. עברו גDAL חלון שונים (משתנה לפי אופי התמונה):

ו. חישוב תగובה הפינה של הפיקסלים עברו חלון בגודל זה ע"פ הנוסחה של ממדנו, ונրמול לגודל החלון:

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

c. עברו כל פיקסל מציאה של התגובה המקסימלית;

d. בחירת כל התשובות מעל לסף מסוימים (סף ייחסי ביחס לתגובה המקסימלית על התמונה);

e. סינון הפיקסלים עבורו מתקבל מקסימום לוקאלי בתגובה הפינה;

f. מציאת האינדקסים של הפינות שנמצאו.

b. חישוב דסקריפטורים – מציאת חלונות מאפיינים // מקבל תמונה ורשימת נקודות (קצוות), ומוחזר רשימה של חלונות מאפיינים עבורו כל פינה:

a. חישוב קירוב לגראדיאנט בפיקסל ע"פ גונני האפור של התמונה;

b. חישוב הזווית של הגראדיאנט;

c. לקיחת חלון בגודל  $44 \times 44$  מסביב לפינה זו;

d. סיבוב החלון כדי לישר את הגראדיאנט כלפי מעלה;

e. לקיחת חלון בגודל  $32 \times 32$ ;

f. הורדת שתי רמות בפירמידה;

g. נרמול ע"פ ממוצע וסטיית תקן.

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

עבור כל חלון מתוך אחת:

a. מציאת שני חלונותاه הכי דומים (במרחק אוקלידי) בתמונה השנייה;

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

c. סינון ומיון מערכיה הפינות לפי התאמות.

ד. // מקבל שני רישימות קצוות מסודרות לפי זוגות מתאימים, מוחזר מטריצת יישור מיטבית (רנדומלי)

עבור מספר איטרציות:

a. הגרלה של 4 התאמות רנדומליות;

b. חישוב מטריצת ההומוגרפיה לפי 4 ההתאמות המוגרלות;

c. בדיקה כמה מההתאמות האחרות, המרחק בין הציפייה למשורה ע"פ מטריצת ההומוגרפיה שמצאנו קטן מסף כלשהו.

הזרות המטריצת עבורה היו התאמות רבות יותר עם מרחק קטן מהסף.

ה'. חישוב מטריצה הומוגרפית מ-4 נקודות // מקבל 4 זוגות נקודות, מוחזר מטריצת יישור  
אנחנו מփשים 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 מירושרת**  
עבור כל פיקסל בתמונה המטריצה:

a. מחשבים את מיקום המקור ע"פ המטריצה הפוכה של מטריצת היישור;

b. מחשבים את ערך הצבעים של הפיקסלים בינהם נפלנו, ע"פ משקל השטח הכלוא המרוחק מהם  
(בילינארי):

a. מחשבים את כל האופציות לשילובי עיגול כלפי מעלה ועיגול כלפי מטה בכל קואדרינטה,

b. דוגמים את הפיקסל המעורר.

c. משקלים לפי מכפלת המרחוקים מהעיגולים ההופכים לאלו שלנו.

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

a. טעינת ערוץ השקיפות של התמונה עם הרזולוציה הגבוהה;

b. יישור שלו ע"פ מטריצת היישור שמצאנו;

c. קביעת המסכה כמעבר סף של ערוץ השקיפות המישר;

d. סכימה ביןארית ע"פ המסכה הבינארית הנ"ל של שתי התמונות: התמונה עם הרזולוציה הגבוהה,  
שיישרנו ע"פ היישור שמצאנו, והתמונה בעלת הרזולוציה הנמוכה.

ח'. **מציאת התאמה בין שתי תמונות // מקבל שתי תמונות ממזוגות בהתאם לקסימלית**  
(rndom)

i. עבור כל תמונה:

.i. טשטוש גאוסיאני;

.ii. מציאת פינות;

.iii. חלונות מאפיינים לכל פינה;

.ii. מציאת התאמות בין חלונות מסוימים התמונה;

.iii. RANSAC לממציאת מטריצת היישור הטובה ביותר;

.iv. הדבקה של התמונה המירושרת לתמונה עם הרזולוציה הנמוכה.

### 3. פרטי מימוש

כרגע, השתמשתי ב-NumPy ו-OpenCV, מושם שהם מאוד נוחים, מאוד פופולריים, ועם דוקומנטציה נחרת.

#### א. חילוץ פיצ'רים – זיהוי צמות ע"פ אלוי Harris

בחorthy להשתמש בקרוב של הא里斯 ע"י הפונקציה שתוארה לעיל באלאג'רים ולא ע"י הע"ע של מטריצת הנגזרות החלקיות מרמה שנייה, עקב העובדה הקירוב קל בהרבה לחישוב ועשה עבודה מעולמת על התמונות שניתנו.

لتמונה הראשונה גדלי החלונות היו [5,11,21,41], לתמונה השנייה [5,11], בלבד. סף לתמונה הראשונה – 0.1, ולתמונה השנייה – 0.3. מקסימום לוקאלי ע"פ חלון בגודל  $3 \times 3$  (רוק מי שיש להם פאה או קדקך מסוית).

מומש עצמאית.

אחרי המון סבבים עם תוצאות מוזרות למדי, גיליתי שהפתרון הוא כמו בתרגילים הקודמים – המהה ל- int32.

#### ב'. חישוב דיסקריפטורים – מציאת חלונות מאפיינים

גודל 44 לחלון לפני היטוב כדי שבכל מקרה אחרי היטוב יהיה חלון 32 × 32 מהקטע.

הוורדה רמה בפירמידה מספרית OpenCV (מיימנו ידנית כבר בתרגיל 3). כל השאר מומש ידנית.

#### ג'. מציאת התאמות בין דסקריפטורים

لتמונה הראשונה סף יחס נמוך יחסית (ברון יותר) – 0.6, ולתמונה השנייה סף יחס גבוה – 0.9. בדיעבד מסתבר שבטעות לא שמתינו כלל סףראשוני ושמתי סף יחס בלבד, מה שהביא אליו התאמות תמוות מהשו, אבל ה-RANSAC פיצה על זה בכל מקרה.

מומש ידנית.

#### ד'. RANSAC

לא כל פעם הייתה מבוסט מהתוצאה אז לפעים הרצתישוב... אז יש תמונות שזכו לפחות או יותר 10000 הרצאות. סף המרחק המקסימלי המותר היה 5 לשתי התמונות. מומש ידנית.

#### ה'. חישוב מטריצת הומוגרפיה מ-4 נקודות

השתמשתי בפונקציה המוכנה של OpenCV. לשימם לב שהסתנدرט שלהם לגביהם ו- $\text{w}$  והוא 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. בכל מקרה תיארתי לעיל את הקווים הכלליים של האלגוריתם.

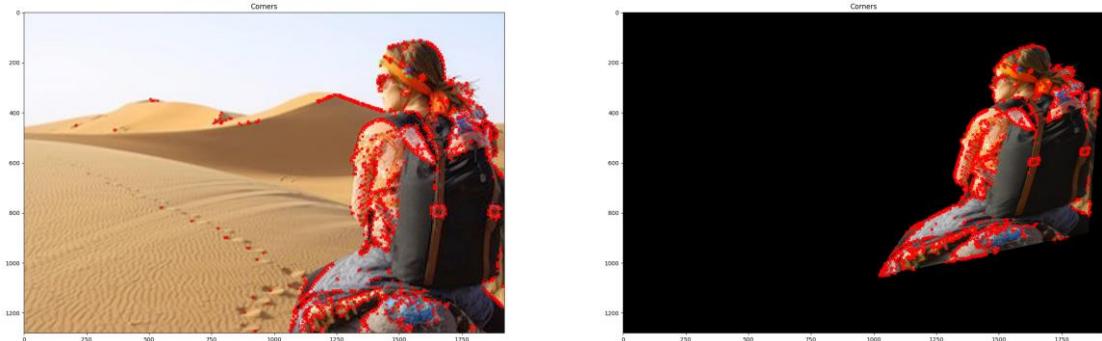
#### ז'. הדבקת שתי התמונות

בחזרתי לעשות את המסכה כסף של עורך השקיפות כדי להימנע מהקו השחור בקריאת הקובץ כ-**RGB** רגיל. ערבול פירמידה לא עוזר בכך הדר הגובה של הקו השחור הזה. ערבול פירמידה טוב באזורי דומים. הסף הוא בערך 240 (מתוך 255). מומש ידנית.

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

### 4. תוצאות ויזואליות

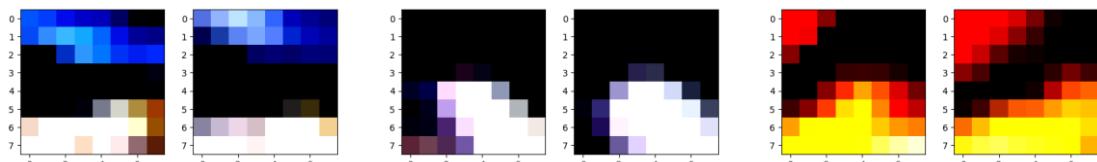
#### א'. חילוץ פיצ'רים – זיהוי קצוות ע"פ אלגי Harris



במבט ראשון נראה שאלה המון פינוט" שלאו דוקא נכונות. מהתבוננות בפاطץ'ים (להלן) נראה שם אכן מספיק מעניינות ומציקות יחס, ופושט ציריך וולוציה גבואה מספיק כדי לשים לב לפיניותו שם.

כמו"כ, מctrע הזיהוי של המסגרת של האובייקט בתמונה עם הרזולוציה הגבוהה, אבל במקרה לנסות לסנן אותן ידנית, ניתן למציאת התאמות לטפל בהן.

#### ג'. מציאת התאמות בין דסקריפטורים דוגמאות לזוגות מתאימים:

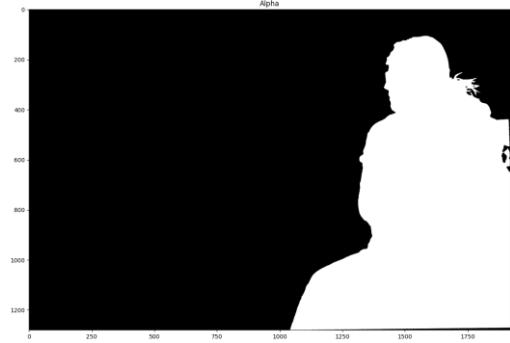


(כל הצבעים מנורמלים, אז החלוניות לא מופיעים בצבעים הללו במקור)



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

#### ד. הדבקת שתי תמונות



(אין פה משהו 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 מסורתיים להתחמת ויזור תמונות (כן אני יודע שזה נשמע ניסוח של בינה מלאכותית, מבטיח שימוש כתבתי את זה בעצמי, זו באמת התהוושה שלי).

אחד הדברים המעניינים הוא שאע"פ שבديעד גילייתי שהזון אי דיקום במימוש שלי (האריס לפיקור ולא לפיקוס) מצליח נזרות סובל מרמה שנייה כמו בIMPLEMENTATION של OpenCV, היישוב גרדיאנט לא לפיקור אלא רק על 4 הפיקסלים הסמכים (סובל לוקח 6), סף התאמת יחסי בלבד בלבד סוף קבוע) והזון מקומות שהזו יכולים לשפר מאוד את זמן הריצה ואת איות התחמות, בעצם אלגוריתם ה-RANSAC חיכה על כל אלו והצליח להביא לתוצאות מרשימות ע"פ שהקלט אליו לא היה מיטבי.

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

**תודה רבה !**