בס"ד

# Image Processing - Exercise 4

Noam Aburbeh, noam.aburbeh, 208883538

## מבוא

בתרגיל זה אנחנו צריכים להלביש קטע של תמונה על תמונה אחרת, וכדי להלביש אותה בצורה הטובה ביותר אנחנו משתמשים בטכניקות של מציאת התאמה בין תמונות כדי להתאים את קטע התמונה לתמונה הגדולה כדי שההלבשה תהיה חלקה, ושלא נשים לב. על ידי מציאת נקודות עניין ומציאת טרנספורמציה בין התמונות.

## אלגוריתם

1. קריאת התמונות
2. הפיכת התמונות לאפורות
3. הפעלת אלגוריתם SIFT על התמונות האפורות ויצירת דיסקריפטורים לכל נקודת עניין שהאלגוריתם מצא.
4. מציאת התאמה בין נקודות על ידי הדיסקריפטורים שמצאנו, על ידי המרחק האוקלידי בין הנקודה שלנו לנקודה הכי קרובה, לעומת המרחק בין הנקודה הקרובה לנקודה הקרובה שאחריה. מה שנקרא 2 nearest neighbor.
5. נעביר עוד סף על גבי הנקודות שמצאנו כדי לראות שאכן המרחק בין השכן הראשון לבין שני השכנים האחרים, באמת קטן במיוחד
6. הפיכת הנקודות לצורה של NUMPY
7. הפעלת אלגוריתם RANSAC כדי להוציא outliers ושיביא את המטריצה M המייצגת את הטרנספורמציה הטובה ביותר בין כל הנקודות
8. הפעלת הטרנספורמציה על התמונה והגדרת הממדים שלה לפי התמונה הראשונה
9. יצירת מסיכה באמצעות הערוץ השקיפות שקיים בקובץ PNG
10. מחיקת כל המקומות בתמונה בעלת האיכות הנמוכה איפה שיש את התמונה באיכות גבוהה
11. חיבור שתי התמונות יחד.

רוב מוחלט של התרגיל נעשה על ידי החבילה של OPENCV, אפרט את המימוש של הפונקציות.

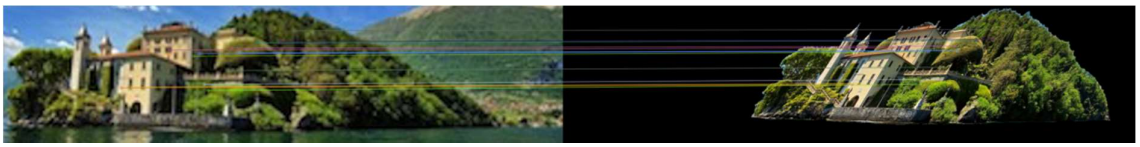
קריאת הנתיב נעשתה על ידי הפונקציה imread, בתמונה באיכות הגבוהה יש את הסימן IMREAD_UNCHANGED כדי שישמור גם את הערוץ השקיפות שיש בקובץ PNG.

נפעיל את אלגוריתם SIFT על התמונות שמצאנו, נזכיר בקצרה איך הוא עובד, בהתחלה הוא מוצא מקסימום מקומי בפירמידת DOG (החסרת גאוסיין מעצמו בסקלות שונות), אחרי מציאת הנקודות הללו, נחשב את הסקלה של הנקודה ואחר כך נחשב את הגרדיאנט. לבסוף נייצר דיסקריפטור עבור כל נקודת מפתח על ידי התחשבות בגדלים ובכיווני שיפוע התמונה המקומיים באזור סביב נקודת המפתח.

אחרי הגדרת נקודות עניין ויצירת הדיסקריפטורים שלהם, ננסה למצוא התאמה בין הנקודות, ואת זה נעשה על ידי הפונקציה flann.knnMatch שעושה את המרחק nearest neighbor כלומר, הוא מחשב את המרחק האוקלידי בין הנקודה שלנו לנקודה הקרובה ביותר שלה, ואז לנקודה הזאת מוצא את

הנקודה הבאה הקרובה ביותר, אם המרחק בין הראשון לשני גדול נוכל להבין כי אכן הנקודה שלנו מתאימה לנקודה הראשונה, הארגומנטים של הפונקציה הם הדיסקריפטורים של שתי התמונות, וגם K=2 שאומר שנבחר עד השכן השני הקרוב ביותר. בחירה זו נלמדה בשיעור שהיא מספיק טובה ויכולה למצוא התאמות טובות. הפלט של הפונקציה הוא טאפל של שתי הנקודות הקרובות ביותר.

אחר כך נעביר רף כדי לסנן עוד יותר תוצאות, ונעשה כך, אם המרחק בין הראשון לשני קטן פי 0.8 מהמרחק בין השכן הראשון לשני, אז נכניס אותם לתוצאה הסופית שלנו. מספר זה נבחר על ידי ניסוי וטעיה ואכן עם ערכים אחרים יצא קצת טעות. למשל בדוגמא למטה נוכל לראות כאשר נבחר סף של 0.9

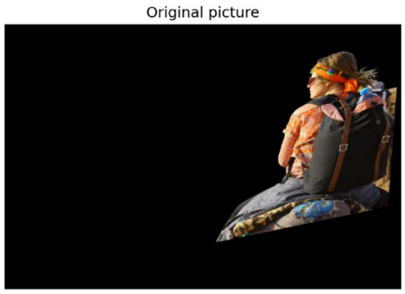


בתמונה כאן מוצגות כל נקודות העניין שמצאנו אחרי כל הדברים שהפעלנו, ניתן לראות שאכן יש התאמה בין הדברים

הפיכת הנקודות לצורה של NUMPYעל ידי הפונקציה np.float32

נמצא את הטרנספורמציה על ידי הפונקציה findHomography שמקבלת כקלט את נקודות העניין שמצאנו משתי התמנות אחרי ההתאמה ואלגוריתם רובסטי, במקרה זה בחרתי את RANSAC כמו שלמדנו בכיתה, ואת הסף 5.0 שנותן את הסטייה האפשרית. נזכיר כי RANSAC במקרה שלנו לוקח נקודות רנדומלית כדי למצוא טרנספורמציה ואז בודק כמה נקודות מסכימות עם זה עד כדי סטייה של 5, והטרנספורמציה שמסכימה הכי הרבה היא הפלט שיוצר בתור מטריצה M, בנוסף יוצא גם פלט מסיכה שמראה את כל הנקודות שהאלגוריתם מצא כנכונות (inliers) כמסכה בינרית.

נפעיל את הטרנספורמציה שמצאנו על ידי הפונקציה warpPerspective שמקבלת את התמונה שרוצים להפעיל עליה, ואת המטריצה, ואת הממדים הסופיים שאנו רוצים לתמונה.


ניתן לראות את התוצאה של הטרנספורמציה על התמונה באיכות הגבוהה, שהיא אכן דומה לתמונה בעל איכות נמוכה.

בס"ד

ניצור מסיכה עם ערוץ השקיפות של התמונה באיכות הגבוהה, ערוץ השקיפות משמש אותנו ליצירת מסיכה מכיוון שכל מקום שהתמונה היא שחורה לגמרי כלומר אין בה אינפורמציה אזי הוא יהיה שחור ובשאר יש ערך כלשהו, ואנחנו נעביר ערך סף ככה שמעל הערך הזה הוא יהיה לבן ומתחת שחור. ניתן לראות את התוצאה מימין. בחרתי את הערך סף עשר על ידי ניסוי וטעייה, שורה אחר כך נהפוך הכל, ונעשה את הבפנים שחור ואת הבחוץ לבן.

נגדיר לפי המסכה את שאר התמונה באיכות נמוכה לשחורה והפוך בתמונה האיכותית, ונחבר בין התמונות על ידי הפונקציה add ונחזיר את התמונה

## תוצאות סופיות

בתמונת המדבר ניתן לראות פס שחור בין התמונה באיכות הגבוהה לבין התמונה באיכות הנמוכה, כאשר עושים זום אין ניתן לראות כי זה ממש פס של שניים שלושה פיקסלים, כלומר זה נכנס לתוך הטווח שנתנו בפונקציה של RANSAC ולכן יכול להיות שהטרנספורמציה לא התאימה בדיוק אלא סטתה בשניים שלושה פיקסלים ולכן רואים את הפס השחור, איך בכללי ניתן לראות כי זה משתלב טוב ונכנס טוב בתמונה.

בתמונת האגם, ניתן לראות בעיה בשני המגדלים שאין שם טרנספורמציה מדויקת, לא כל כך הצלחתי להבין למה, ניסיתי לשחק עם ערכי הסף שהיו לי באלגוריתם אבל זה לא שיפר את התוצאות, במיוחד שבשאר המקומות נראה שזה נעשה בצורה טובה. אפשר לנסות להעריך כי מציאת נקודת העניין במקומות אלו הייתה קשה יותר מכיוון שגם המגדל עצמו חלק וגם במקומות של השינויים הרקע גם כן יותר לבן בגלל העננים מאחורה ולכן קשה למצוא בדיוק את הנקודה המתאימה.

# סיכום

נהניתי מאוד לעשות את התרגיל! כמו פעמים קודמות היישום של הטכניקות שלמדנו בכיתה זה מעניין ומלמד, היה נחמד לראות גם את התהליך ולבדוק בעיניים מה התוצאות שיוצאות. לראות שבאמת האלגוריתם הצליח למצוא לא מעט התאמות בין נקודות גם אחרי הרבה מבחני סף שהיה צריך לעבור היה מלמד. בנוסף היה מגניב לראות ששינוי ערכי הסף באמת שינה לגמרי את התוצאות והיה צריך למצוא את הערך הנכון.