

ראייה ממוחשבת

תאריך: _____8/1/2021_____

שם סטודנט: _____אביב כספי_____

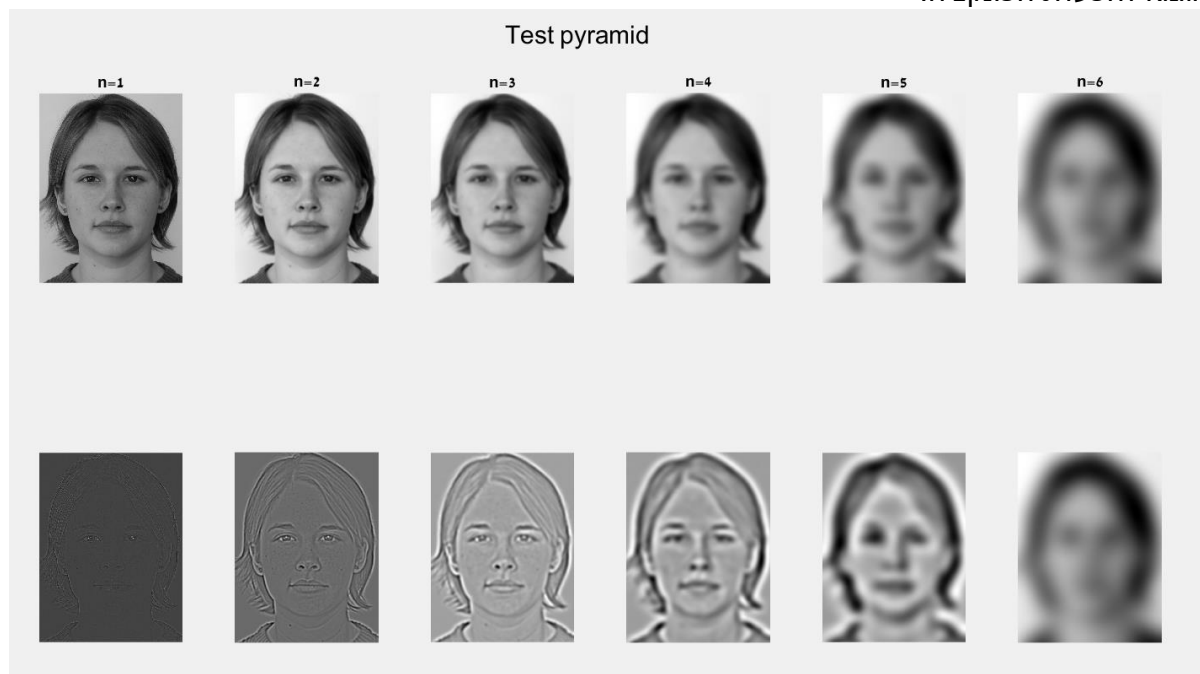
מספר סטודנט: _____311136691_____

מייל: _____avivcaspi@campus.technion.ac.il_____

מספר גיליון: _____3_____

שאלה 1

a. בסעיף זה בניתי פירמידה לפלסיאנית, לפי התיאור שמפורט בתרגיל, כאשר תחילה חישבתי את הפירמידה הגאוסית ולאחר מכן השתמשתי בכל שתי דרגות של הפירמידה הזאת על מנת לחשב דרגה אחת של הפירמידה הפלסיאנית. דוגמא להפעלת הפונקציה:



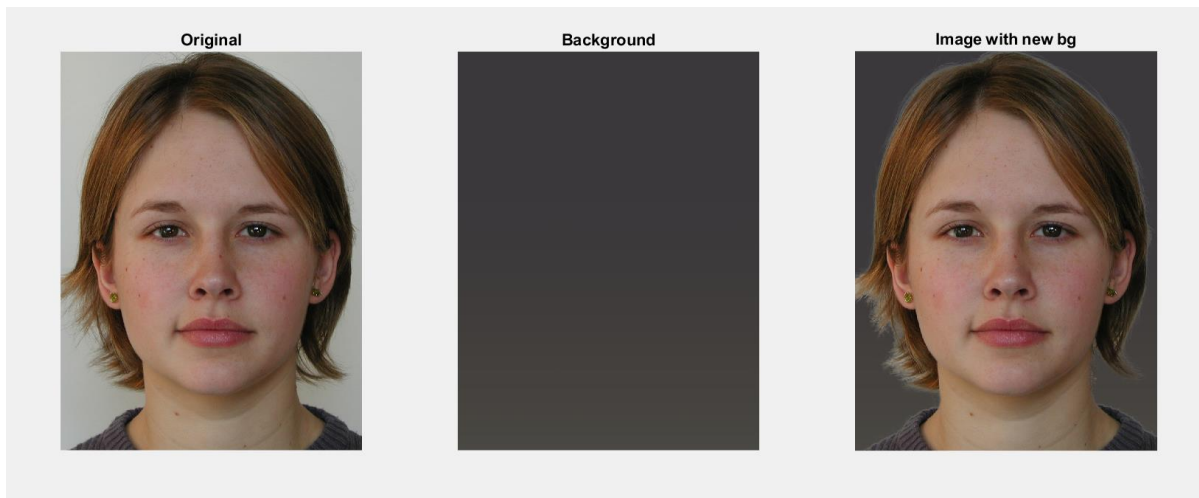
ניתן לראות בשורה הראשונה את הפירמידה הגאוסית, ובשורה השנייה את הפירמידה הפלסיאנית.

b. ניתן לראות כי הדרגה האחרונה של שתי הפירמידות זהה כמו שתכננו. כעת בניתי פונקציה שמקבלת את הפירמידה הפלסיאנית ומשחזרת את התמונה המקורית על ידי סכימה של כל דרגות הפירמידה כמו שראינו בתרגול. חשוב להגיד, כי אין לנו צורך לבצע *Upsampling* בין הדרגות השונות, מפני שלא ביצענו *down sampling* כאשר בנינו את הפירמידה.

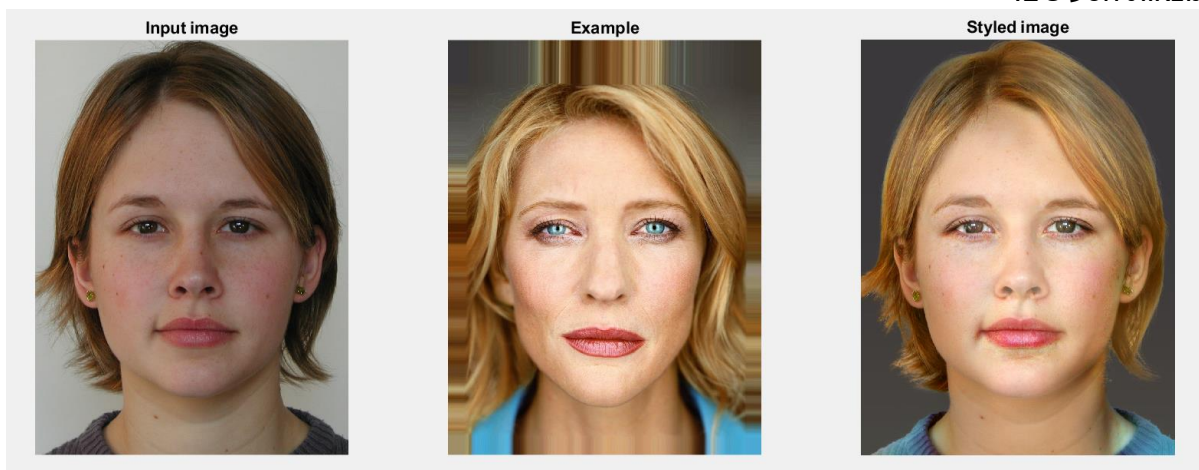


ניתן לראות שהשחזור שקיבלנו כמעט מושלם לגמרי, וכי השגיאה שקיבלנו הינה מינימלית ובעצם אפסית. במקרה בו היינו משתמשים ב*down sampling* בזמן בניית הפירמידה, היינו מקבלים שחזור פחות טוב מאשר השחזור שקיבלנו עכשיו. זה נובע מכך שהפעלת הגאוסין אינה מושלמת בכל דרגה, ולכן בעצם בזמן ביצוע הטשטוש עם

הגאוסייין ואז ביצוע *down sampling* אנחנו בעצם מוחקים מידע, שלא ניתן לשחזר בצורה מלאה. לכן השחזור שנקבל יהיה פחות מדויק מהשחזור שקיבלנו עכשיו.
 c. בסעיף זה, יצרתי פונקציה אשר מחליפה את רקע התמונה, ברקע אחר, על ידי שימוש במסכה שמצורפת לתרגיל.
 דוגמא להרצה:



d-f. מימוש סעיפים אלה נמצא בקובץ הפונקציה *styleTransfer*.
 בסעיף זה חישבתי את האנרגיה ואת ה *gain* כמו שצויין בתרגיל, ואיחדתי את הפירמידות שחישבנו לתמונות כמו שתואר.
 תוצאות הסעיפים:



ניתן לראות כי התמונה מצד שמאל קיבלה את המאפיינים של תמונת הדוגמא האמצעית.
 כאשר מצד ימין ניתן לראות את התוצאה הסופית.

g. כעת הפעלתי את האלגוריתם על תמונות נוספות, אלה התוצאות שקיבלתי:

Input image



Example 16



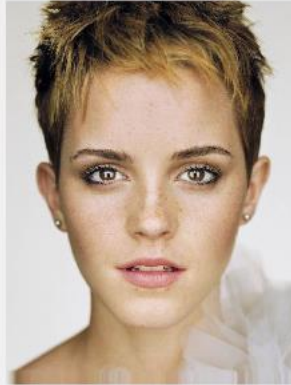
Styled image



Input image

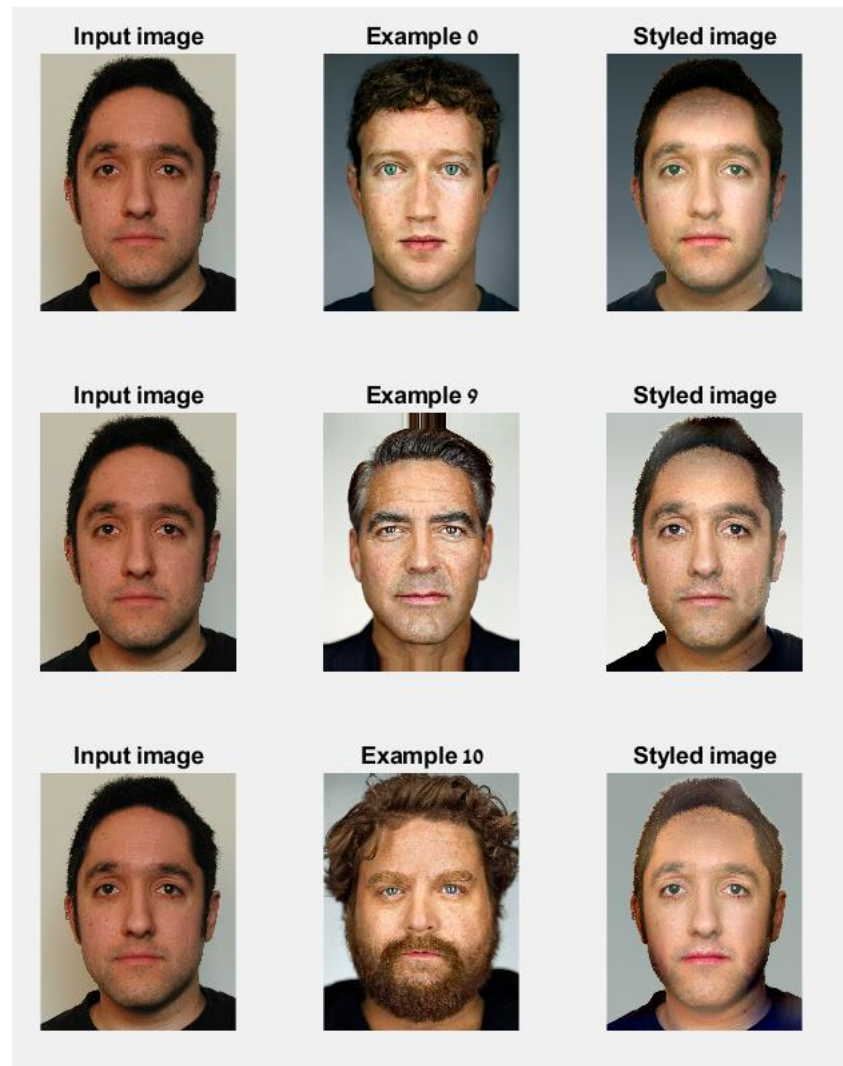


Example 21



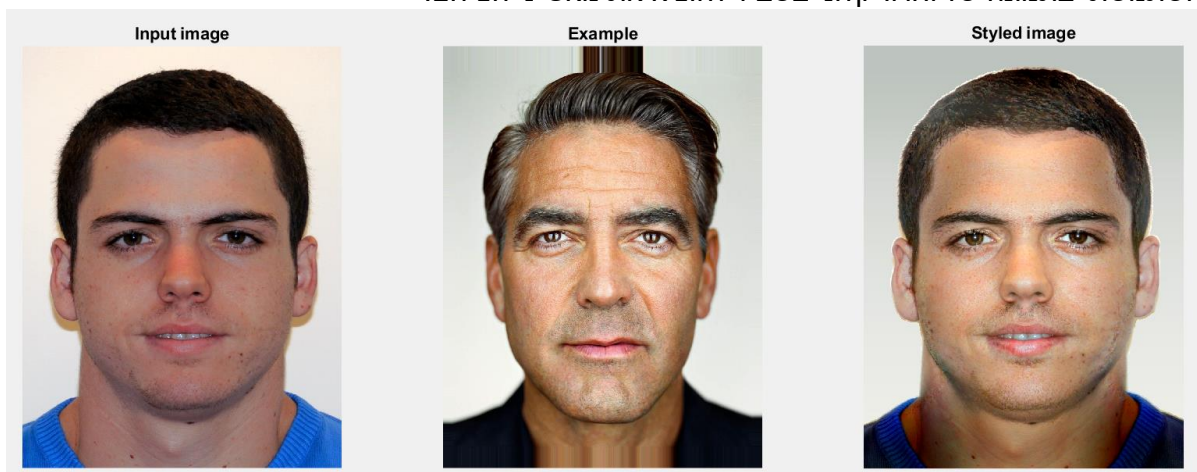
Styled image





בכל הדוגמאות ניתן לראות כי המאפיינים של התמונה האמצעית, עברו לתמונה המקורית בצורה יפה.

h. כעת הפעלתי את האלגוריתם על תמונה שלי, כאשר בניתי לה מסכה כדי לשנות את הרקע, והשתמשתי בתמונה של ג'ורג' קלוני בשביל להוציא את מאפייני הצילום.

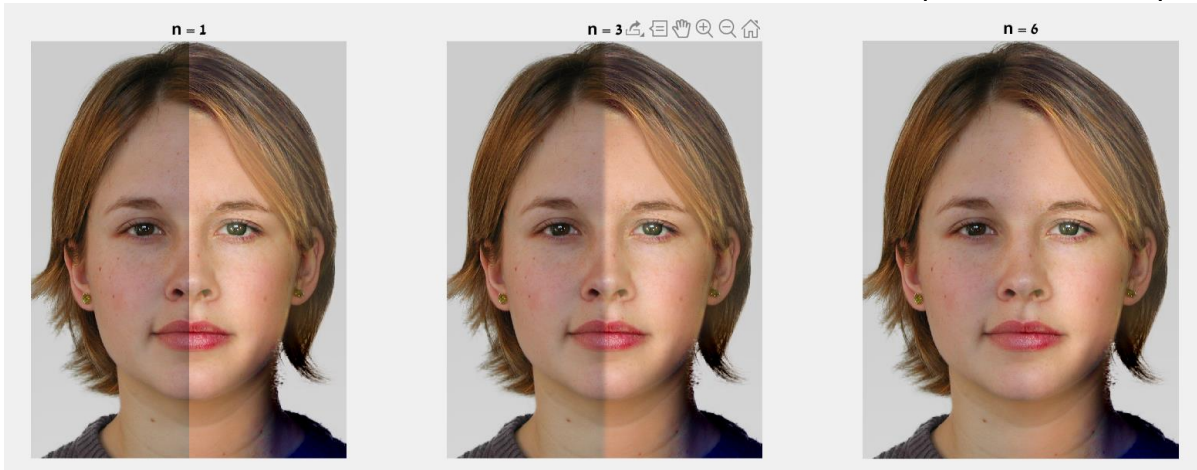


גם בדוגמא זו ניתן לראות כי המאפיינים של התמונה האמצעית עברו לתמונה שלי.

i. בסעיף זה מימשתי את אלגוריתם *pyramid blending* שלמדנו בתרגול. כחלק מהאלגוריתם, תחילה חישבתי את פירמידות הלפליין של כל אחת מהתמונות (המקורית והתמונה עם הסטייל החדש).

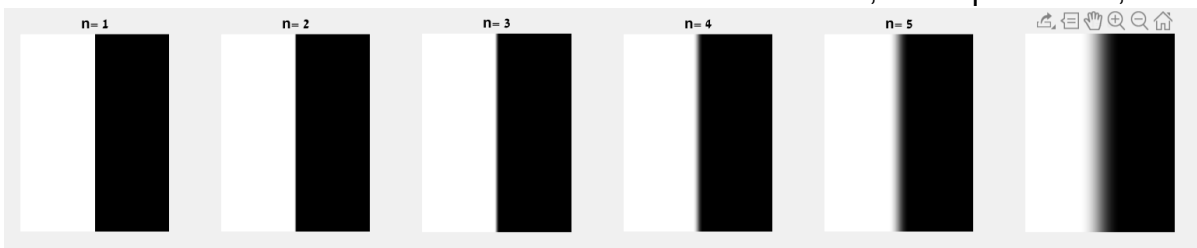
לאחר מכן, בניתי מסכה בצורת מדרגה, כאשר ערכה הוא 1 עד האמצע האופקי של התמונה, ולאחר מכן הוא 0.

לקחתי את המסכה הזאת וחישבתי את הפירמידה הגאוסית שלו. כעת כשיש בידי את כל הפירמידות הנ"ל, ביצעתי איחוד של הפירמידות בעזרת המסכה, לכל אחת מהדרגות של הפירמידות בנפרד, על מנת לקבל פירמידה חדשה, אשר משלבת בין התמונות. לאחר ביצוע איחוד זה, השתמשתי בפונקציית השחזור שבניתי בסעיפים קודמים על מנת לקבל את התמונה הסופית מהפירמידה. להלן התוצאות עבור עומק פירמידה משתנה:



ניתן לראות כי ככל שעומק הפירמידה שנשתמש לביצוע המיזוג גדל, כך הקו המפריד בין התמונות נראה יותר ויותר טוב וטבעי.

השיפור בהפרדה, נובע מכך שככל שנכנס לדרגה עמוקה יותר ויותר בפירמידה הגאוסית של המסכה, ההפרדה בין הצדדים, יותר ויותר מטושטשת.

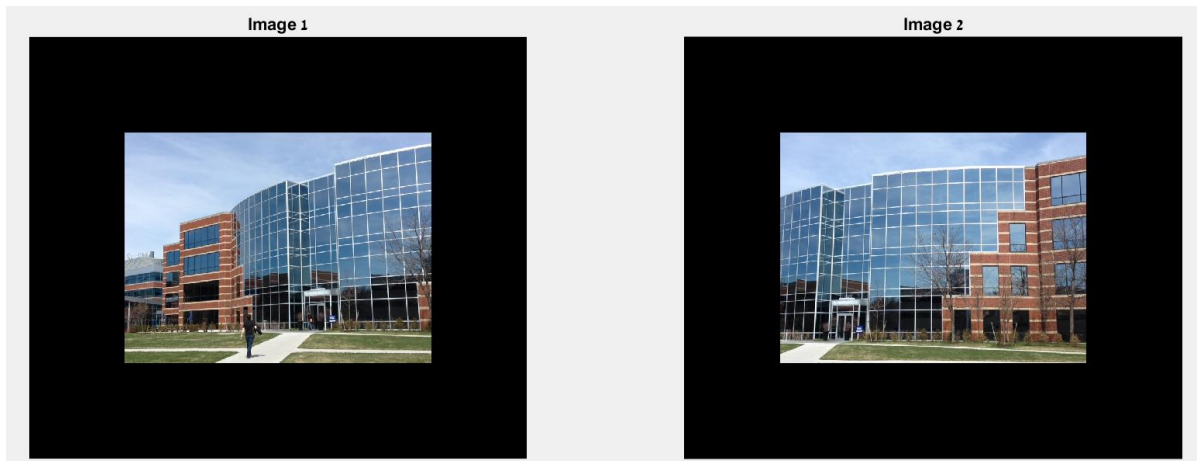


נוכל לראות כאן את הפירמידה הגאוסית של המסכה, ואכן ניתן לראות כי בעומקים 1-4, ההפרדה מאד ברורה בין הצדדים, וניתן לראות את הפס המפריד.

אך עבור העומקים 5-6 ניתן לראות כי הפס המפריד נהיה יותר מטושטש ופחות ברור. זה גורם לכך שההפרדה בעומק הפירמידה הלפלסיאנית של התמונה הממוזגת הופכת לחלקה יותר.

שאלה 2

1. התמונות שבחרתי לשאלה זו:

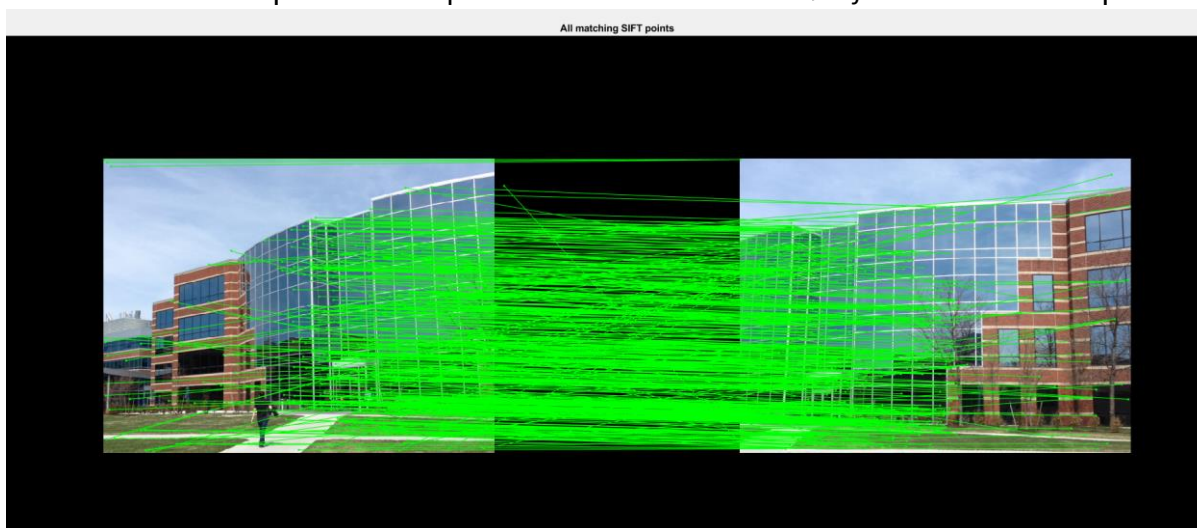


אציין, כי כבר בשלב זה בחרתי להוסיף *padding* לתמונות, על מנת להקל על פעולת ה*warping* בסעיפים מתקדמים יותר.

הוספת ה*padding* לא משפיעה על אף סעיף עד אז.

2-3. בשני הסעיפים האלה, השתמשתי בפונקציית *SIFT* המצורפת לתרגיל, על מנת למצוא את הנק' החשובות בכל תמונה ואת התיאור שלהם.

לאחר מכן השתמשתי ב*siftmatch* על מנת למצוא את הזוגות של הנק' המתאימות בין שתי התמונות.



ניתן לראות בקווים ירוקים את הקישורים בין כל זוג נק'.

ניתן לראות בתמונה זו כי יש זוגות נק', שאינן מתאימות אך האלגוריתם חושב שהן כן מתאימות.

על מנת לסנן נק' אלה, נשתמש באלגוריתם *RANSAC* בסעיפים הבאים.

4. בסעיף זה מימשנו את האלגוריתם שלמדנו בכיתה על מנת למצוא את הטרנספורמציה שתמיר לנו שתי נק' בתמונה אחת, למיקום שלהם בתמונה השנייה.

על מנת לחשב את הטרנספורמציה, עלינו לדעת לפחות 4 זוגות של נק' מתאימות.

נראה מדוע זה נכון:

בתרגול ראינו את המשוואות הבאות:

$$\begin{bmatrix} u'_i \\ v'_i \\ w'_i \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad \begin{aligned} x'_i &\rightarrow \frac{u'_i}{w'_i} \\ y'_i &\rightarrow \frac{v'_i}{w'_i} \end{aligned}$$

The equations we get are

$$\begin{aligned} x'_i &= \frac{ax_i + by_i + c}{gx_i + hy_i + 1} \\ y'_i &= \frac{dx_i + ey_i + f}{gx_i + hy_i + 1} \end{aligned} \Rightarrow \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

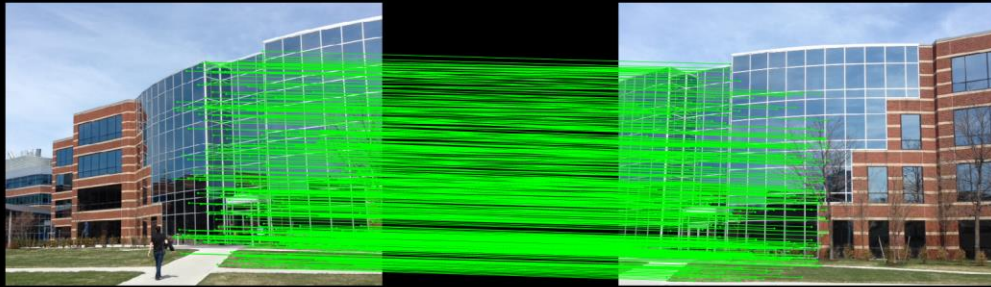
How do we solve? SVD!

משוואות אלה מתארות כיצד הפעלת טרנספורמציה, יכולה להביא אותנו מנק' אחת לשנייה המתאימה לה. כאשר הטרנספורמציה המתאימה היא המטריצה העליונה מהצורה 3×3 . כעת נשים לב, כי ניתן לשנות את צורת הכתיבה של המשוואות ולקבל משוואה הומוגנית, כאשר כל זוג נק' יוצרות 2 שורות במטריצה A , ומטריצת הטרנספורמציה שלנו הופכת לוקטור באורך 9 שמכיל את ערכיו. על מנת לפתור מערכת משוואות כזאת, המכילה 8 משתנים בלתי תלויים, אנחנו צריכים מטריצה בעלת לפחות 8 שורות. כלומר, אם ידוע לנו שכל זוג נק' יוצר 2 שורות במטריצה, נצטרך לפחות 4 זוגות של נק' על מנת לחשב את המטריצה H של הטרנספורמציה. הפונקציה שמימשת, מקבלת את זוגות הנק', ומחשבת את הטרנספורמציה הנכונה על ידי שימוש ב SVD כמו שראינו בעבר באלגוריתם ל $camera\ calibration$. הבעיה שנרצה לפתור היא:

$$\operatorname{argmin}_h |Ah| = 0$$

וראינו כבר בעבר, כי הפתרון לבעיה זו, הוא הוקטור העצמי המתאים לערך העצמי הקטן ביותר של A ועל מנת למצוא את הוקטור הזה נשתמש בפירוק SVD .

5. כעת מימשת את אלגוריתם $RANSAC$, אשר מקבל את זוגות הנק' שמצאנו ב $sift$, ומחשב את הטרנספורמציה H המתאימה למספר מקסימלי של זוגות. ניתן לראות את תוצאות האלגוריתם בתמונה הבאה:



כעת, ניתן לראות כי האלגוריתם שלנו, הצליח לסנן את כל הזוגות שנראו לנו לא מתאימים, ונותרנו עם קווים שנראים כמעט זהים אחד לשני.

6. כעת, מימשתי פונקציה אשר מקבלת את התמונה הימנית ומפעילה עליה את הטרנספורמציה שמצאתי בסעיף הקודם.

בתחילת החישוב, הוספתי *padding* מימין לתמונה המקורית, על מנת שתוצאת העיוות תהיה כולה בתוך התמונה שנקבל, ללא חיתוכים.

לאחר מכן, השתמשתי באלגוריתם *backwarp mapping*, אשר מחשב את הטרנספורמציה ההופכית H^{-1} ומשתמש בה כדי למצוא לכל קוארדינאטה בתמונת המעוותת את הקוארדינאטות שמתאימות להן בתמונה המקורית.

יש לשים לב, כי יש להפעיל אינטרפולציה על האינדקסים שמתקבלים, מפני שהם אינם שלמים, ונרצה לדעת את ערך התמונה באינדקסים השלמים.

תוצאת העיוות של התמונה הימנית הינה:

Building 2 warped



8. כעת מימשתי פונקציה אשר מבצעת איחוד לתמונות, כאשר בחיתוך בין התמונות הצגתי רק את אחת מהן:

Final result



נשים לב, כי בגלל שהשתמשתי בטרנספורמציה שמצאתי על התמונה הימנית, בעצם העברתי את האינדקסים שלה, למערכת הצירים של התמונה השמאלית.
כלומר, כל שיש לעשות עם התמונה השמאלית הוא להוסיף אותה לתמונה המעוותת כמו שהיא.