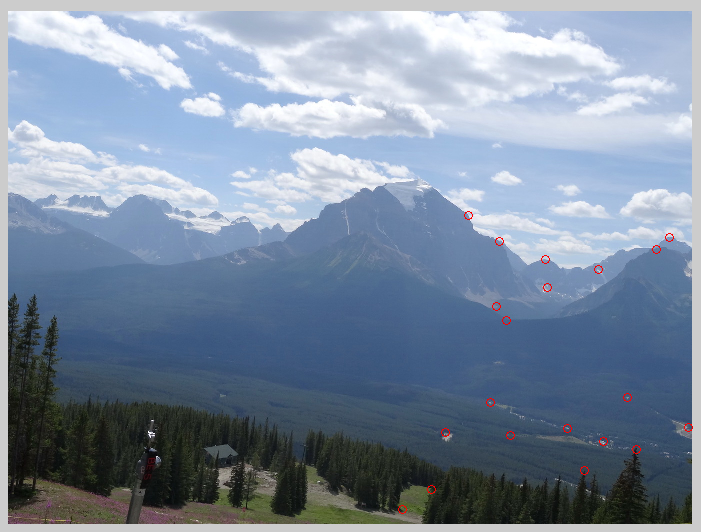
**מגישים: אור היבנטריגר 301020632**

**הלל שרטר 300636404**

נטען את הקובץ matches\_perfact.mat ונראה כי כל הנקודות תואמות:

Src:

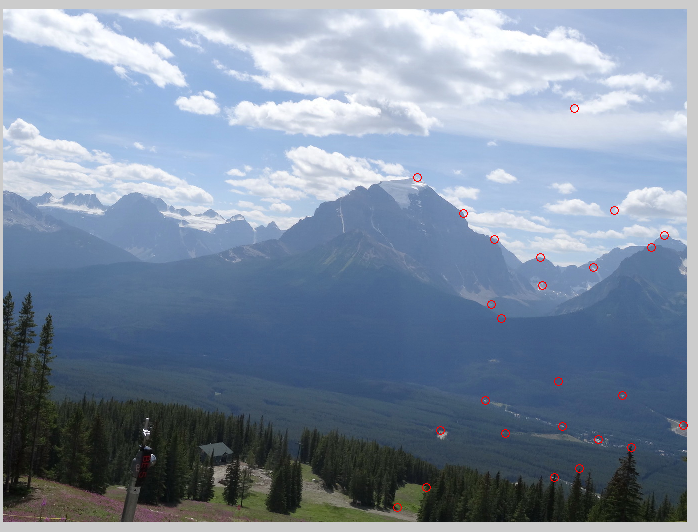


Dst:



נטען את הקובץ matches.mat ונראה כי לא כל הנקודות תואמות (ישנם outliers):

Src:



Dst:



**חלק א' – חישוב הומוגרפיה:**

1. עבור התמרה פרויקטיבית נבנה מערכת משוואות מהצורה . מטריצת ההתמרה H ניתנת לתיאור בצורה הבאה – 

\* קביעת האיבר האחרון ל-1 היא שרירותית. ניתן לקבוע כל ערך שנרצה. למעשה, פתרון ע"י SVD (כפי שיוסבר בהמשך) מאלץ שסכום ריבועי האיברים יהיה 1.

סה"כ קיימים 8 נעלמים ולכן יש צורך ב-4 נקודות כדי לפתור את מערכת המשוואות.

נסתכל על מערכת המשוואות -



נכתוב את משוואות המערכת ונקבל -



נציג זאת בצורה מטריצית ונקבל –

את המטריצה H , שמיוצגת ע"י x בייצוג CS, ניתן למצוא ע"י לכסון A ובחירת הוקטור העצמי שמתאים לערך העצמי הנמוך ביותר. ניתן לבצע זאת בקלות ע"י SVD.

2. מצורפת הפונקציה.

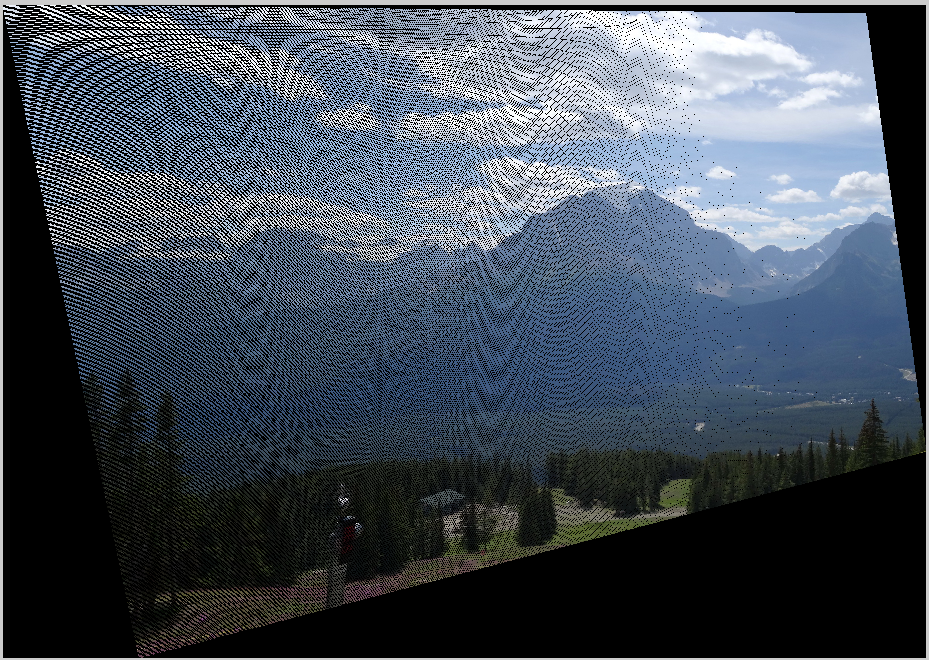
3. התוצאה היא:

H = -0.0011 -0.0002 0.9999

-0.0000 -0.0011 0.0126

-0.0000 -0.0000 -0.0008

4. להלן תמונת המקור לאחר התמרה פרוייקטיבית, בהתאם למקדמים מסעיף 3:



5. הבעיות המרכזיות של ForwardMapping הן:

* תמונת המקור ותמונת היעד אינן בהכרח בעלות גודל זהה. לכן, נקודות סמוכות בתמונת הקלט יכולות להיות רחוקות יותר בתמונת הפלט. הדבר יגרום להיווצרות חורים שחורים בתמונת הפלט.
* הקואורדינטה  אינה בהכרח קואורדינטה שלמה.

הבעיה שבאה לידי ביטוי בתמונה היא שנקודות סמוכות בתמונת הקלט אינן בהכרח סמוכות בתמונת היעד. כיוון שביצענו התמרה רק על תמונת המקור ניתן להבחין בתופעה בצורה ברורה – חורים שחורים שהתגלו בתמונה ואליהם לא מופה אף פיקסל מתמונת היעד.

6. כעת מקדמי ההתמרה הינם:

H = 0.0006 0.0001 -0.6078

0.0006 0.0005 -0.7941

0.0000 0.0000 -0.0010

כעת ישנם outliers. כתוצאה מכך אנחנו מקבלים התמרה לא נכונה. כאשר ננסה למצוא להיכן יותמרו ארבעת קודקודי תמונת היעד נקבל ערכים שאינם הגיוניים (מספרים גדולים מאוד שלא ניתן לייצגם במחשב). לפיכך, לא הצלחנו לקבל תוצאה בגלל בעיות חישוביות.

**חלק ב' – התמודדות עם Outliers:**

7. מצורפת הפונקציה.

8. מצורפת הפונקציה.

9. ע"פ מה שנלמד בהרצאה נסמן:

W - אחוז ה-inliers.

p - ההסתברות לבחירת inliers בלבד.

n - מספר מינימלי של נקודות כדי לשערך.

במקרה זה, מספר האיטרציות k נתון על ידי:

נתון כי קיימות 30 נקודות התאמה, כאשר 80% מהן נכונות.

עבור מקרה בו נרצה להבטיח הצלחה בשיעור של 90%, מספר האיטרציות יהיה:



עבור מקרה בו נרצה להבטיח הצלחה בשיעור של 99% מספר האיטרציות יהיה:



על מנת לכסות את כל האפשרויות, עלינו לבצע איטרציות. נתונים אלו ממחישים את היתרון שיש ל-RANSAC על פני פתרון נאיבי.

10. טענו את הקובץ match.mat וחישבנו את מקדמי ההתמרה באמצעות הפונקצייה compute\_homography.m. מקדמי ההתמרה שהתקבלו הינם:

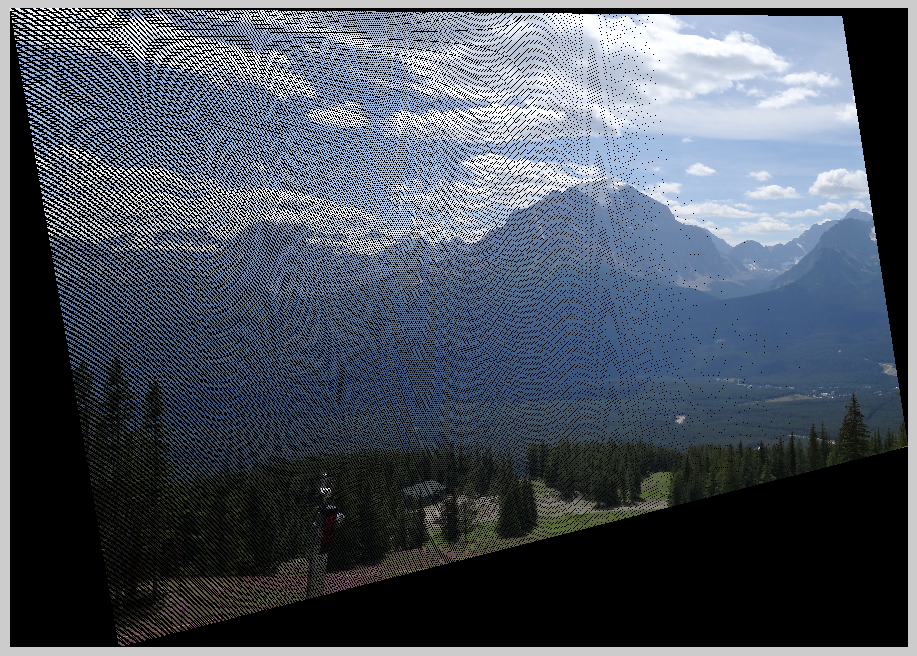
H = -0.0011 -0.0002 1.0000

-0.0000 -0.0010 -0.0018

-0.0000 -0.0000 -0.0008

ניתן לראות שהתקבלו מקדמים שקרובים למקדמים שהתקבלו ללא נוכחות outliers.

להלן התמונה שהתקבלה לאחר השימוש במקדמי ההתמרה שהתקבלו:



השוואה לתוצאה שהתקבלה בסעיפים 4-6 עבור הנקודות ללא ה-outliers: ניתן לראות שתמונות הפלט מאוד דומות ובקושי ניתן להבדיל ביניהן. המסקנה היא שהשימוש ב-RANSAC הוכיח את עצמו, והשערוך התבצע כנדרש.

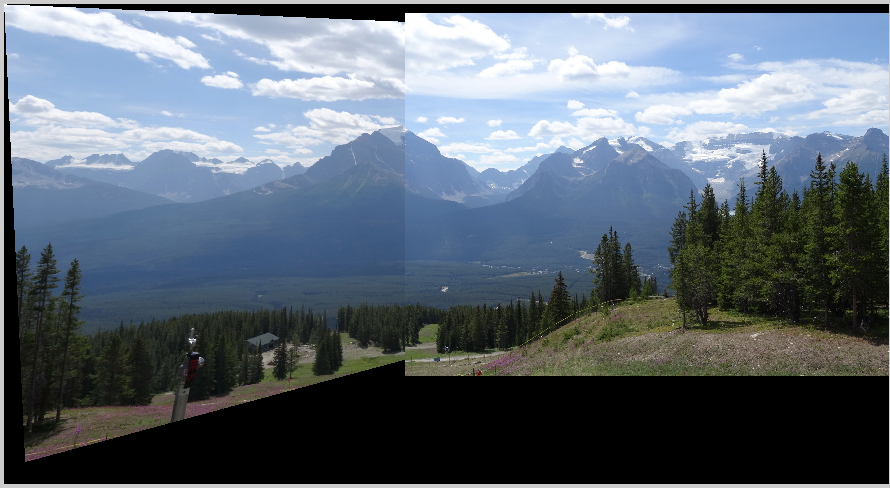
**חלק ג' – בניית פנורמה:**

11. להלן תמונת המקור לאחר Backward Mapping :



ניתן לראות שכעת קיבלנו תמונה מושלמת ללא חורים וללא הבעיות שדיברנו עליהם בחלק ב' בהקשר Forward Mapping. ניתן לראות שמנח התמונה וגודלה זהה לסעיף 10, רק שכעת התמונה מופיעה ללא חורים.

12. מצורפת הפונקציה.

13. הפעלנו את הפונקצייה panorama על התמונות הנתונות בתרגיל עם הפרמטרים הנתונים. להלן תמונת הפנורמה שהתקבלה:

ניתן לראות שהפונקציה עובדת כנדרש.

14. כעת נשתמש בזוג תמונות שצילמנו לבניית הפנורמה באמצעות פונקציית panorama.

נציג את תמונות המקור והיעד יחד עם נקודות ההתאמה שסומנו:

Src\_test:



Dst\_test:



ניתן לראות שבחרנו ב- 25 זוגות של נקודות תואמות, מתוכן 4 זוגות של נקודות שאינן תואמות (נק' מספר 10, 12, 16, 23).

להלן תוצאת הפנורמה שהתקבלה:

