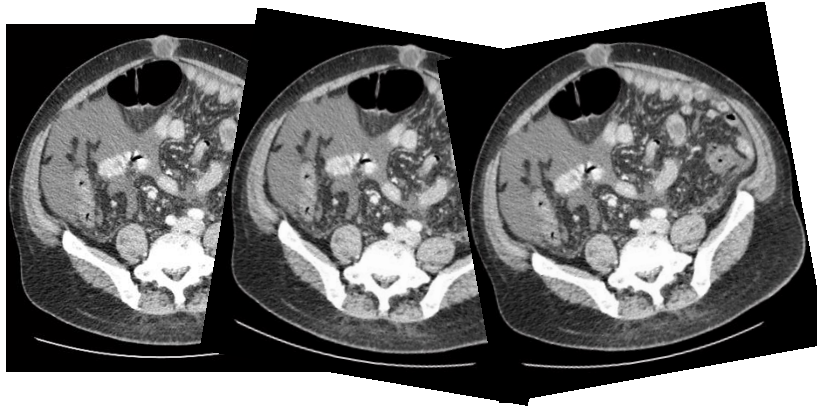


עבודה מספר 2 בקורס עיבוד תמונות רפואיות 67705

נושא: רגיסטרציה

פרופ' ליאו יוסקוביץ - סמסטר א' תשע"ז

עורך התרגיל: גיא מדן gmedan@cs.huji.ac.il



כללי

1. הגשת התרגיל הינה ביחידים.
2. לעבודה זו שני חלקים בשני מועדי הגשה: החלק הראשון יוגש עד **5/1/2017** והחלק השני עד **19/1/2017**. את הציון תקבלו רק לאחר ההגשה השנייה.
3. גרסת מטלב המתאימה לביצוע התרגיל – **2015b** או מאוחר יותר.
4. בכל הסעיפים יש להניח כי שתי התמונות אשר עוברות רגיסטרציה הינן באותו גודל בפיקסלים וכל פיקסל בעל אותם מימדים בכל הצירים.
5. על מנת לאפשר תרגול יעיל של החומר הנלמד, סופקו לכם פונקציות עזר תחת התיקה `utils`. כדי להשתמש בהן יש לקרוא ל- `utils.functionName` כאשר תיקיית העבודה במטלב היא התיקייה המכילה את `utils`.
6. את הקוד שלכם יש לכתוב אך ורק תחת התיקיות `+part1,+part2`. כדי לקרוא לפונקציות שכתבתם יש לקרוא ל- `part1.myFunctionName`.
7. יש להקפיד על הגשה מסודרת (10% מהציון בתרגיל): כותרות לגרפים, תיאורים ויחידות לצירים, גדלים נכונים בסקאלות הצירים.
8. בעת הצגת תמונות במטלב, יש להקפיד להוסיף את הפקודה `axis xy equal` הגורמת לצירים להתחיל מהפינה השמאלית תחתונה, כך שיתאימו לקונבנציה בפונקציות העזר.
9. עליכם להגיש דו"ח ובו תשובות לשאלות המילוליות ו/או עם תמונות בסעיפים בהם התבקשתם לכך. לנוחותכם שאלות מילוליות וסעיפים בהם נדרש צירוף תמונה מסומנים בקו תחתון. בנוסף עליכם להגיש את הקוד שכתבתם.

10. מבנה ההגשה: לכל אחד משני החלקים יש להגיש קובץ zip אחד. שם הקובץ יהיה `ex2_part#_zip` כאשר # הוא מספר החלק ו-`<id>` הוא תעודת הזהות כולל ספרת ביקורת. בתוך כל קובץ יימצא הדו"ח תחת השם `ex2_part#.pdf` והקוד תחת התיקה `#part`. אין להגיש את הנתונים מתיקיית `data` או את הקוד מתיקיית `+utils`. הדו"ח יכול את שם המגיש, `login`, ות"ז.

11. את הקבצים יש להעלות דרך המודל.

12. 10% מהניקוד מבוסס על הגשה מסודרת, לרבות קוד קריא ומתועד.

מבוא

באופן כללי, בין שתי סריקות של אותו פציינט בזמנים שונים אין התאמה טובה. בין שני מועדי הסריקה הפציינט נשם, זז, ואולי גם בילה כמה חודשים בבית. למרות זאת, לעיתים קרובות נרצה שהתמונות יהיו בהתאמה מושלמת, לצרכי השוואה ומעקב, או לטובת טיפול מונחה הדמייה (הקרנה, החדרת מחט) בנקודה הנכונה ע"פ תוכנית טיפול שגובשה על גבי תמונת הסריקה הראשונה. אם האיבר שמעניין אותנו הוא קשיח, כמו המוח לדוגמה, אזי ניתן למצוא את התנועה הזו בעזרת רגיסטרציה ריגידית.

רגיסטרציה ריגידית היא תהליך של מציאת סיבוב והזזה בין זוג תמונות. לפיכך בתמונות תלת מימדיות, כמו MRI או CT למשל, רגיסטרציה ריגידית מאופיינת בשישה פרמטרים: שלושה פרמטרי הזזה, ושלוש זוויות סיבוב. סט של הזזות וסיבובים כאלו נקרא "טרנספורמציה". בתהליך רגיסטרציה אנו מחפשים את הטרנספורמציה בין שתי תמונות, ולעיתים אנו נדרשים גם להפעיל את הטרנספורמציה על אחת התמונות על מנת ליישר אותה כלפי התמונה השנייה.

אלגוריתמים למציאת רגיסטרציה ריגידית מתחלקים באופן כללי לשני סוגים: מבוססי תמונה, או מבוססי נקודות עניין. באלגוריתמים מבוססי נקודות עניין אנו תחילה מזוהים אזורים קטנים בשתי התמונות שמתאימים האחד לשני, כלומר מייצגים כנראה את אותה נקודה באיבר הנסרק. ברגע שיש לנו מספיק התאמות כאלה ניתן לחשב את הטרנספורמציה. באלגוריתמים מבוססי תמונה עושים את ההיפך: תחילה מנחשים טרנספורמציות כלשהן, מפעילים אותן על התמונות (כלומר מיישרים תמונה אחת כלפי השנייה), ואז מודדים את הדמיון בין התמונות אחרי היישור, ובוחרים את הטרנספורמציה בעלת הדמיון החזק ביותר. לעיתים קרובות אלגוריתמים כאלו הם איטרטיביים, כלומר הם משתמשים בטרנספורמציה טובה על מנת למצוא טרנספורמציה טובה יותר וחוזר חלילה. אנו נקדיש חלק מהתרגיל לכל סוג. בנוסף, שאלה אחת תוקדש לרגיסטרציה של תמונות CT במישור ראדון, גישה המשלבת עקרונות משתי השיטות.

חלק 1. אלגוריתמים מבוססי נקודות עניין ומבוססי תמונה (45%) - להגשה עד 5/1/2017 23:59

מטלה א': בחלק זה נמצא רגיסטרציה ריגידית בין שתי תמונות אופטיות דו ממדיות.

רקע: אלגוריתמים מבוססי נקודות עניין מוצאים רגיסטרציה בין שתי תמונות במספר שלבים אוטומטיים:

- זיהוי נקודות עניין.
- מציאת 'התאמות' בין נקודות עניין בשתי התמונות.
- סינון outliers.
- חישוב הרגיסטרציה מתוך ההתאמות הנכונות.

בתרגיל זה נניח שנקודות העניין וההתאמות ביניהן בשתי התמונות כבר נבחרו (ושהן עלולות לכלול outliers) ונכתוב פונקציה אשר משתמשת בהן כדי לחשב את הרגיסטרציה הריגידית בין שתי התמונות. נשתמש בסכמת RANSAC (Random Sample Consensus) כדי לסנן את ה-Outliers בחישוב הטרנספורמציה ולסמן מי הם ה-inlier ומי הם ה-outliers.

1. (3%) טענו למטלב את התמונות brain_moving, brain_fixed ע"י טעינת brain.mat. השתמשו בפונקציה `utils.getPoints('no_outliers')` `[fixedPoints, movingPoints] =` שתחזיר נקודות דו מימדיות במערכת צירי התמונה בתור שתי מטריצות בעלות N שורות ושתי עמודות כל אחת, כך שההתאמות הן בין שורות מתאימות בשתי המטריצות. הציגו את ההתאמות בעזרת צילום מסך של subplot עם שתי התמונות בעזרת פונקציית עזר שתכתבו (לא לשכוח את axis xy). העזרו בפונקציה `text` על מנת להציג את מס' הנקודה ליד סימן המציין את מיקומה. מהו מספר ההתאמות השגויות (כלומר, שנראה בבירור מהתמונות שצמד הנקודות לא מתאר את אותו מבנה אנטומי בתמונה)?

2. (5%) ממשו את

`function rigidReg = calcPointBasedReg(fixedPoints, movingPoints)` שמחשבת את פרמטרי הרגיסטרציה מתוך התאמות נתונות. `rigidReg` היא מטריצה 3×3 . הערכים יהיו כאלה שאם מפעילים את הטרנספורמציה על `movingPoints`, מקבלים נקודות קרובות ככל האפשר לנקודות `fixedPoints` במובן Least-Squares. העזרו בהסבר [כאן](#) על מנת לבצע את החישוב ע"י SVD. התוצאה שלכם צריכה לקיים בקירוב:

$$\begin{bmatrix} \text{movingPoints} & \text{ones}(N,1) \end{bmatrix} * \text{rigidReg} == \begin{bmatrix} \text{fixedPoints} & \text{ones}(N,1) \end{bmatrix}$$

כאשר N הוא מספר הנקודות. שימו לב שמבנה המטריצה `rigidReg` מורכב ממטריצת סיבוב עם וקטור הזזה:

$$\begin{bmatrix} R(2 \times 2) & 0; \\ T(1 \times 2) & 1 \end{bmatrix}$$

3. (5%) ממשו את

`function d_pixel = calcDist(fixedPoints, movingPoints, rigidReg)` המחשבת את המרחק של כל נקודה שעברה טרנספורמציה מהנקודה אליה היא אמורה להתאים בפיקסלים. חשבו את ה **Root Mean Square Error** מדד השגיאה של הרגיסטרציה מתוך הפלט של הפונקציה שכתבתם (הפונקציה מחזירה וקטור באורך N, אך בסעיף זה יש להחזיר ערך סקלרי יחיד ביחידות של פיקסלים). מהו ערך השגיאה בפיקסלים עבור ההתאמות שנטענו בסעיף א'?

4. (3%) ממשו פונקציה שטוענת את שתי התמונות ואת הנקודות שבחרתם, מחשבת את הרגיסטרציה ביניהן, מפעילה את הטרנספורמציה על התמונה `moving` ומציגה אותן זו על גבי זו באופן שיאפשר לראות שכעת הן נמצאות ברגיסטרציה זו ביחס לזו. רמז: ניתן להציג את אחת התמונות עם `edges` בלבד, או כל אחת בצבע שונה בעזרת `imshowpair`. צפו לדו"ח את שתי התמונות ברגיסטרציה.

5. (3%) כעת טענו את `[fixedPoints, movingPoints] = utils.getPoints('with_outliers')` וחזרו על סעיפים 1 ו-4. האם התמונות ברגיסטרציה? מהו ערך השגיאה? מדוע זה כך?

6. (8%) קראו את התיעוד והדוגמאות של `utils.ransac.ransac1`. עליכם להשתמש בה על מנת לכתוב פונקציה

```
[rigidReg, inliers] = calcRobustPointBasedReg( fixedPoints, movingPoints)
```

שתצליח לחשב את הטרנספורמציה למרות הימצאות אחוז מסויים של `outliers` לא ידועים מראש ברשימת ההתאמות. יש לספק ל-`utils.ransac.ransac1` שתי פונקציות אותן היא תפעיל על הנתונים: אחת לחישוב טרנספורמציה מתוך תת-קבוצה נתונה של התאמות, ואחת לחישוב השגיאה של הטרנספורמציה על כל אחת מההתאמות המקוריות. העזרו בפונקציות שכתבתם בסעיפים הקודמים ובאופרטור `@` של מטלב.

7. (3%) חזרו על סעיף 5 אך השתמשו בפונקציה הרובוסטית לחשוב הרגיסטרציה. הדגישו בעזרת צבעים שונים/צורות שונות אילו מהנקודות הן `inliers` ואילו הן `outliers`.

8. בדקו את עצמכם ע"י יצירת סט נקודות מלאכותיות, ביצוע טרנספורמציה ידועה עליהן לקבלת סט נוסף של נקודות, הוספת רעשים אקראיים קטנים וגם התאמות שגויות שהן `outliers`, ושחזרו הטרנספורמציה המקורית מההתאמות בעזרת הפונקציה שכתבתם. אין צורך להגיש קוד או תשובות בסעיף זה.

פונקציות מטלב שימושיות לחלק זה:

`imagesc, svd, affine2d, imwarp, edge, imshowpair, axis equal xy`

מטלה ב': בחלק זה נשתמש בפונקציה של מטלב לרגיסטרציה מבוססת תמונה ונשווה את תוצאות החישוב לתוצאות מטלה א'. כפי שנכתב לעיל, באלגוריתמים מבוססי תמונה מנחשים תחילה טרנספורמציות כלשהן, מפעילים אותן על התמונות (כלומר מיישרים תמונה אחת כלפי השנייה), ואז מודדים את הדמיון בין התמונות אחרי היישור, ובחרים את הטרנספורמציה בעלת הדמיון החזק ביותר. חלק חשוב בתיאור האלגוריתם הוא נתינת ציון לכל טרנספורמציה אפשרית בהתאם לתמונות.

בתחום עיבוד התמונה נפוצים ציונים ומרחקים רבים, וביניהם `NCC`, `SSD`, `SAD`, `MI` ועוד.

בסעיף זה נבדוק את שני הציונים הבאים:

Normalized cross correlation (NCC)

Mutual Information (MI)

הציון `NCC` משמש במקרים שהתמונות דומות בערכי האפור שלהן, למשל שתי סריקות באותו פרוטוקול סריקה. הציון `MI` משמש במקרים שהתמונות דומות במבנה שלהן אבל שונות מאוד בערכי האפור, למשל שתי סריקות `MRI`, האחת בפרוטוקול `T1` והשנייה בפרוטוקול `T2`.

1. (3%) טענו את שתי התמונות ממטלה א' והשתמשו בפונקציה `imregtform` כדי לחשב את הרגיסטרציה הריגידית

ביניהן. ספציפית, העזרו בקריאה ל-

```
tform = imregtform(A,RA,B,RB,'rigid',optimizer,metric)
```

כאשר RA,RB הם אובייקטים מסוג imref2d, והאובייקטים optimizer,metric מוחזרים ע"י imregconfig('monomodal') המאפשר לבצע רגיסטרציה באמצעות ציון NCC. קראו בעזרה של מטלב כיצד יש להשתמש בכל הנ"ל.

2. (4%) הציגו את שתי התמונות ברגיסטרציה. חשבו את הציון NCC בעזרת הפונקציה normxcorr2 לפני ואחרי הרגיסטרציה. הסבירו את התוצאה.

3. (4%) טענו את התמונות (התלת מימדיות) "T1", "T2_rot_n" ע"י קריאה ל- utils.getScan, למשל utils.getScan('T1'). בצעו רגיסטרציה ריגידית שלהן בעזרת imregconfig('multimodal'). הציגו מספר פרוסות מייצגות מתוך התמונות זו על גבי זו לפני ואחרי רגיסטרציה. הציגו בנוסף את ההיסטוגרמה הדו-מימדית של ערכי האפור בשתי התמונות בעזרת קריאה ל-histcounts2, לפני ואחרי רגיסטרציה. העזרו ב-imagesc כדי להציג את התמונות. ניתן לבצע log על ההיסטוגרמות כדי לשפר את התצוגה. הסבירו את התוצאה. מדוע ההיסטוגרמה אחרי רגיסטרציה נראית כפי שהיא נראית, וכיצד היא שונה מההיסטוגרמה דומה המחושבת על שתי תמונות ברגיסטרציה מאותו modality? מדוע תוכננו הפרוטוקולים T1, T2 כך שההיסטוגרמה המחושבת על סריקות שנמצאות ברגיסטרציה זו עם זו נראית כפי שהצגתם?

4. (4%) ממשו את function nmi = calcNMI(A,B) המחשבת את מדד ה-NMI עבור שתי תמונות תוך שימוש ב-histcounts2. חשבו על דרך אוטומטית לבחור את גבולות ומספר התאים של ההיסטוגרמה בהנתן התמונות. הוסיפו את תוצאות חישוב מדד ה-NMI עבור שתי התמונות מסעיף 3 לפני ואחרי רגיסטרציה.

נספח: במטלב 2014a לא קיימת histcounts2 ולכן ניתן להעזר בפונקציה utils.jointHistogram הלקוחה מפתרון ב"ס לסעיף הבא בתרגיל של השנה שעברה, ולהשתמש בה באופן דומה ל- histcounts2(X,Y,Xedges,Yedges) :

(20%) ממשו את

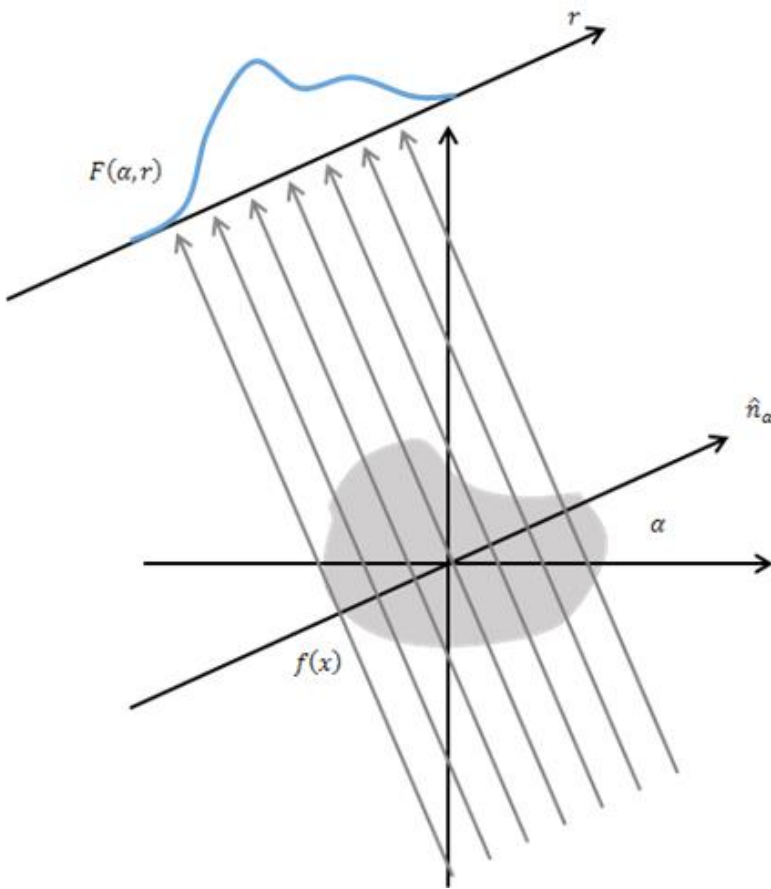
```
function count = jointHistogram(image1, bins1, image2, bins2)
```

שלמדתם בכיתה. מכיוון שחישוב ההיסטוגרמה מבוצע מחדש בכל איטרציה של אלגוריתם הרגיסטרציה, על המימוש להיות יעיל ככל שניתן.

- image# - מטריצה תלת מימדית של ערכי אפור בסריקה. זהו שדה התמונה באובייקט המוחזר מ- utils.getScan, או טרנספורמציה שלה שחושבה בעזרת imwarp.
- bins# - מתאר את גבולות התאים של רמות האפור להיסטוגרמה. למשל: bins1 = [0 50 100 150] מתאר 3 תאים: [0,50), [50,100), [100,150). כל ווקסל שקטן ממש מ-0 או גדול-שווה ל-150 בתמונה 1 לא צריך להספר בהיסטוגרמה המשותפת.
- הניחו שהערכים ב-bins# מונוטוניים עולים ממש ועם מרווחים קבועים (אין צורך לבדוק זאת בקוד).
- על הגודל של הפלט count להיות: length(bins1)-1 שורות, length(bins2)-1 עמודות. בתא ה-ij יופיע מספר המופעים של זוגות ווקסלים כך שהווקסל מתמונה 1 נמצא בקטע החצי סגור [bins1(i), bins1(i+1)) והווקסל מתמונה 2 נמצא בקטע החצי סגור [bins2(j), bins2(j+1)).

חלק 2. אלגוריתם רגיסטרציה במישור ראדון (45%) - להגשה עד 19/1/2017 23:59

בחלק זה נמצא רגיסטרציה ריגידית בין סריקות דו מימדיות במישור ראדון, כלומר מתוך הסינוגרמה של תמונות CT. בסריקת CT של פרוסה אחת, מקור קרני רנטגן מסתובב סביב גוף המטופל ומקליט את טרנספורמצית ראדון, או סינוגרמה, של ערכי הבליעה ברקמה שבין המקור לגלאי. כפי שלמדנו, אלגוריתמים שונים כגון algebraic reconstruction, filtered backpropagation מאפשרים לבצע טרנספורם ראדון הופכי ולשחזר את התמונה שזהו טרנספורם ראדון שלה. לאחר מכן, בין תמונות של אותו מטופל מסריקות שונות ניתן למצוא רגיסטרציה בשיטות אותן בחנו בחלק 1. החיסרון של שיטת דימות זו הוא בחשיפת המטופל לקרינת רנטגן במינונים גבוהים. בחלק זה נתעניין דווקא בביצוע חישובי רגיסטרציה על הסינוגרמה עצמה, וזאת מהטעם הבא: אם נוכל למצוא רגיסטרציה בין סריקות CT תוך שימוש בחלק מזערי מהמידע שבסינוגרמה מהסריקה השנייה, הרי שכמות קרינת הרנטגן המזיקה אליה נחשף המטופל בסריקה השנייה יכולה תיאורטית לרדת באופן משמעותי. מכיוון שאין לנו צורך ברוב המידע מהסריקה השנייה לצורך ביצוע הרגיסטרציה, אין לנו צורך בסריקת המטופל מכל כיוון אפשרי אלא רק ממספר קטן של כיוונים.

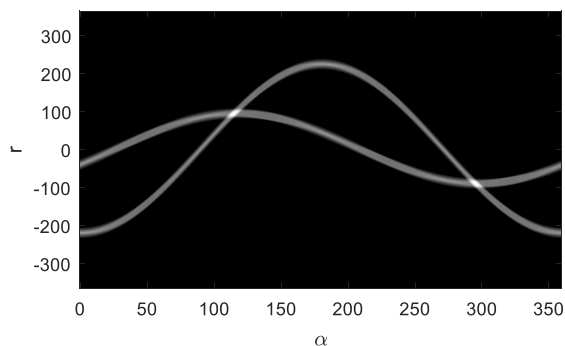
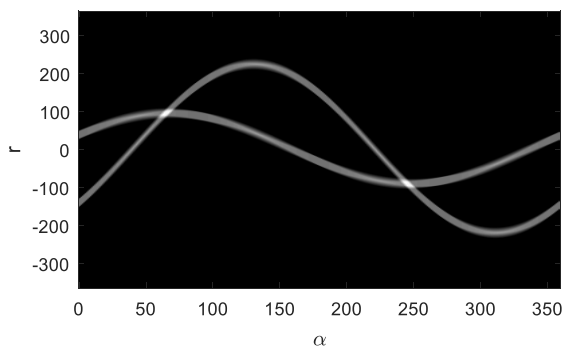
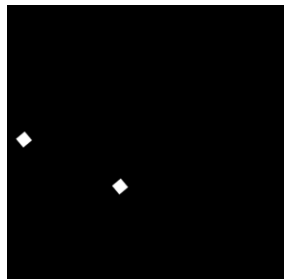
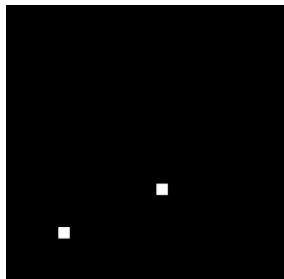


כפי שראיתם בכיתה, טרנספורם ראדון מוגדר ע"י אינטגרל קווי במישור:

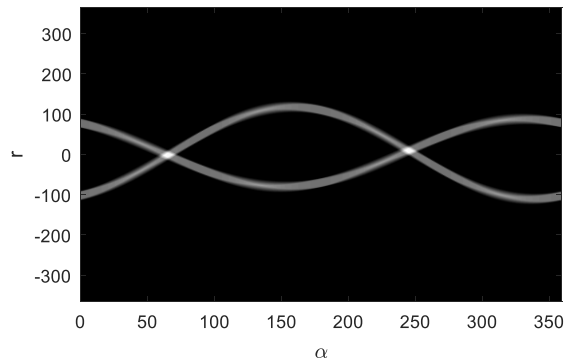
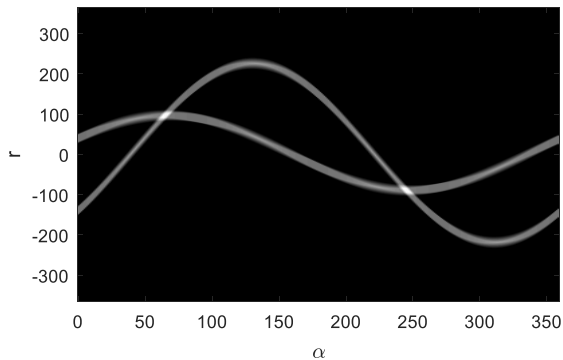
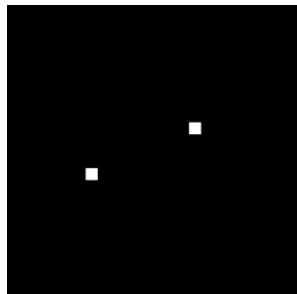
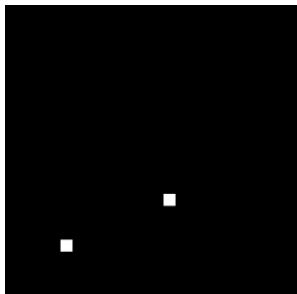
$$F(\alpha, r) = \int_{l(\alpha, r)} f(x) dl$$

כאשר f היא תמונה דו מימדית, F טרנספורם ראדון שלה, x קורדינטה דו מימדית בתמונה, α היא קורדינטת זווית ההיטל של טרנספורם ראדון, r קורדינטת ההיסט של טרנספורם ראדון. \hat{n}_α הוא וקטור הנורמל לכיוון קו ההטלה, dl הוא האינטגרנד על הקו $l(\alpha, r): x \cdot \hat{n}_\alpha = r$.

קיים קשר אנליטי בין טרנספורם ראדון של תמונות f, g שיש ביניהן טרנספורמציה ריגידית. כדי לראות זאת נתבונן בדוגמה הבאה, בה קיים סיבוב בלבד בין שתי התמונות:



בין שתי התמונות בשורה העליונה קיים סיבוב של 50 מעלות, ובין הסינוגרמות שלהן בשורה התחתונה קיים הקשר הבא: אחת היא הזזה ציקלית בציר α (הציר האופקי) של השנייה ב-50 מעלות. כעת נתבונן בדוגמה בה בין שתי התמונות קיים קשר של הזזה בלבד. ניתן לראות שאין הזזה בסינוגרמות בציר α האופקי, אך יש הזזה בציר z האנכי, שגודלה משתנה עם הזווית α :



עבור טרנספורמציה ריגידית המשלבת סיבוב וגם הזזה, כלומר $g(x) = f(R_\theta x + \Delta)$ כאשר $R_\theta \in \mathbb{R}^{2 \times 2}$ היא מטריצת סיבוב בזווית θ נגד כיוון השעון, Δ וקטור הזזה דו מימדי, ניתן להוכיח כי הקשר האנליטי הבא מתקיים בין הסינוגרמות F, G :

$$G(\alpha, r) = F(\alpha + \theta, r - \delta_\alpha)$$

כלומר, ההיסט בציר α הוא בדיוק θ , וגודל ההיסט δ_α בציר r תלוי בזווית ההטלה α דרך השוויון: $\delta_\alpha = \Delta \cdot \hat{n}_\alpha$.
 כעת נתאר אלגוריתם למציאת פרמטרי הרגיסטרציה הריגידית θ, Δ (שלושה פרמטרים סה"כ):

- לכל זווית α בצע:

- בחר את G_α , העמודה המתאימה לזווית הטלה α מתוך הסינוגרמה G
- מצא מתוך כל עמודות F את העמודה $F_{\alpha'}$ עבורה מתקבל ציון קורלציה הטוב ביותר עם G_α , כולל היסט δ של $F_{\alpha'}$ בציר r , כלומר:

$$\alpha', \delta = \operatorname{argmax} \operatorname{CrossCorrelation}(G_\alpha, T_\delta F_{\alpha'})$$

כאשר T_δ אופרטור ההסטה ב- δ .

- שמור את המדידה $(\alpha, \alpha', \delta)$

- חשב היסטוגרמה של הערכים $\alpha - \alpha'$ מתוך סט המדידות ובחר את המקסימום כדי למצוא את θ
- פתור את מערכת המשוואות הלינארית $\{\delta = \Delta \cdot \hat{n}_\alpha\}$ - משוואה אחת לכל מדידה, כאשר הנעלם המופיע בכל המשוואות הוא Δ .

נשים לב לפרטים הבאים:

- אין צורך אמיתי לבצע את התהליך עבור כל זווית α - די לנו ב-2 זוויות שונות על מנת לפתור מערכת משוואות בשני נעלמים. עם זאת, כדי לשפר את הדיוק, ולהתגבר על רעש נומרי ו-outliers, עלינו לקחת מספר גדול יותר של זוויות, אך כמובן שמספר זה לא מתקרב לגודל סט הזוויות הנדרש על מנת לשחזר את התמונה מתוך הסינוגרמה.
- הפחתה זו במספר הזוויות היא המאפשרת את החיסכון בקרינת רנטגן, שכן כל זווית הטלה שלא סרקנו משמעותה הפחתת כמות הקרינה בסריקה השנייה.
- קרוס-קורלציה מבוצעת על וקטורים חד מימדים שהם עמודות מתוך הסינוגרמה, כאשר אחד מהווקטורים מוסט ביחס לאחר כדי למצוא את ההיסט שנותן את הקורלציה הטובה ביותר.
- פתרון מערכת משוואות בה מספר המשוואות גדול ממספר הנעלמים נעשה ע"י מזעור השגיאה הריבועית, כלומר פתרון מערכת המשוואות $Ax = b$ מתקבל ע"י $x^* = \operatorname{argmin} \|Ax - b\|_2^2$ (וקיים לבעיית האופטימיזציה הזו פתרון אנליטי).

א. (30%) ממשו פונקציה

```
function [rotation_ccw_deg, translation_pixels] = radon_register(fixed_sinogram,
fixed_angles_deg, moving_sinogram, moving_angles_deg)
```

כך ש-`fixed_sinogram`, `moving_sinogram` הן סינוגרמות של שתי תמונות שחושבו בעזרת הפונקציה של מלב `radon` על הזוויות `fixed_angles_deg`, `moving_angles_deg` בהתאמה. הפונקציה מחזירה את הסיבוב והזזה שנמצאו ע"י האלגוריתם לרגיסטרציה ריגידית במישור ראדון שתואר למעלה. כאן `fixed` מקבילה ל-`F` ו-`moving` מקבילה ל-`G`.

פונקציות מועילות לחלק זה: `xcorr`, `mldivide`

ב. (10%) ממשו פונקציה

```
function show_radon_registration(img1, rotation_ccw_deg, translation_pixels,  
num_angles)
```

המבצעת את הפעולות הבאות:

- מייצרת מ-`img1` תמונה `img2` באותו גודל ע"י סיבוב נגד כיוון השעון ואז הזזה
- מחשבת את הסינוגרמה של `img1` עבור וקטור של זוויות `0:1:179`
- מחשבת את הסינוגרמה של `img2` עבור וקטור של זוויות באורך `num_angles` ברווחים שווים בתחום $[0, 180)$ (שימו לב שהטלה בזווית 180 מעלות זהה ל-0 מעלות עד כדי היפוך)
- קוראת לפונקציה מסעיף א'
- מבצעת על `img2` טרנספורמציה רגיסטרציה בעזרת התוצאות
- מציגה את שתי התמונות זו על גבי זו (בעזרת `imshowpair`) לפני ואחרי הרגיסטרציה.

מומלץ להעזר בפונקציה `imwarp` כדי לבצע סיבובים/הזזות/רגיסטרציות. כדאי לעבוד עם אובייקטי `imref2d` המאותחלים כך שהסיבוב יהיה סביב מרכז התמונה ולא סביב הפינה השמאלית עליונה כדי שתהיה התאמה לאופן בו פועלת הפונקציה `radon`, ועם אובייקטי `affine2d`.

ג. (5%) הציגו את התוצאות מהסעיף הקודם עבור תמונה המתקבלת מהפונקציה `phantom`, ועבור תמונת CT המצורפת לתרגיל (ממטלה א' סעיף 1 – יש לנרמל את הערכים כך שהאוויר יהיה בעל רמת אפור אפס). הסבירו בקצרה את המימוש שלכם לאלגוריתם המוצע, הרחיבו על שיפורים שהוספתם ככל שישנם (למשל סינון `outliers`).