

פרויקט במערכות נבונות - 236754

---

# ***Surgery Simulation using Image Processing***

דוד פטחוב  
פלג הראל

מנחה  
דוד פלוס

מרצה אחראי  
אהוד ריבלין



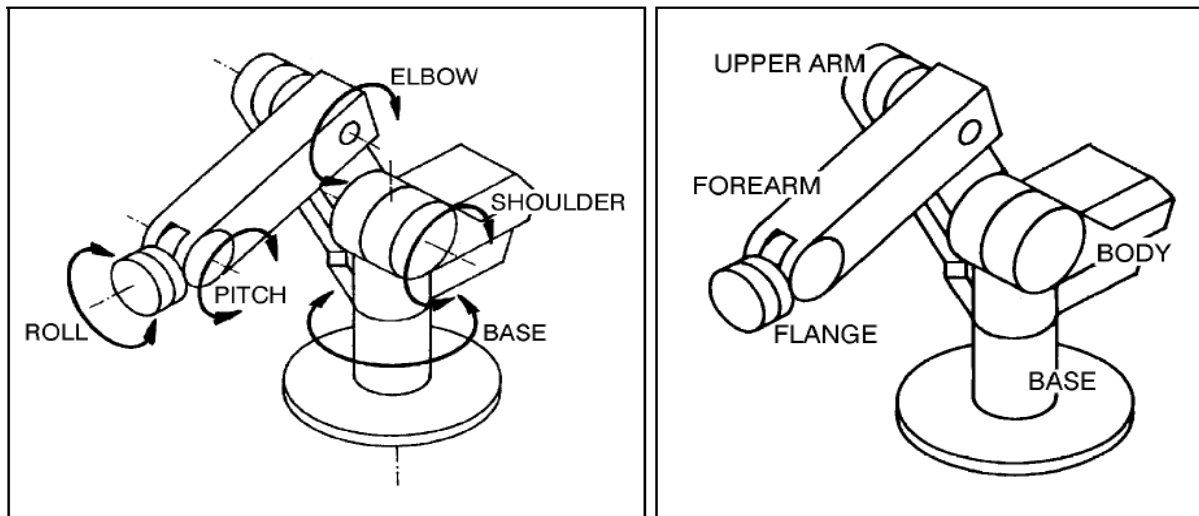
## תוכן עניינים

---

3	תקשורת עם הרובוט .....
4	מעבר מקואורדינטות העולם לזוויות המפרקים של הרובוט (reverse trajectory) .....
4	חישוב זוויות בשביל נקודה יחסית .....
8	חישוב זוויות בשביל נקודה מוחלטת .....
10	הדמיית ניתוח באמצעות עיבוד תמונה .....
10	הקדמה .....
10	רקע .....
11	מטרה .....
11	מימוש .....
11	קליברציה של המצלמה .....
12	תיקון עיוות .....
13	מעבר מנקודה במערכת הצירים של הרובוט לזו של המצלמה .....
13	מעבר ממערכת הצירים של הרובוט למערכת הצירים של לוח השחמט .....
13	מעבר ממערכת הצירים של לוח השחמט למערכת הצירים של המצלמה .....
14	מעבר מנקודה במערכת הצירים של הרובוט לזו של המצלמה .....
14	תהליך בחירת הנקודה .....
16	הבאת הרובוט לנקודה הרצויה בזוויות הנכונות .....
17	ביבליוגרפיה .....

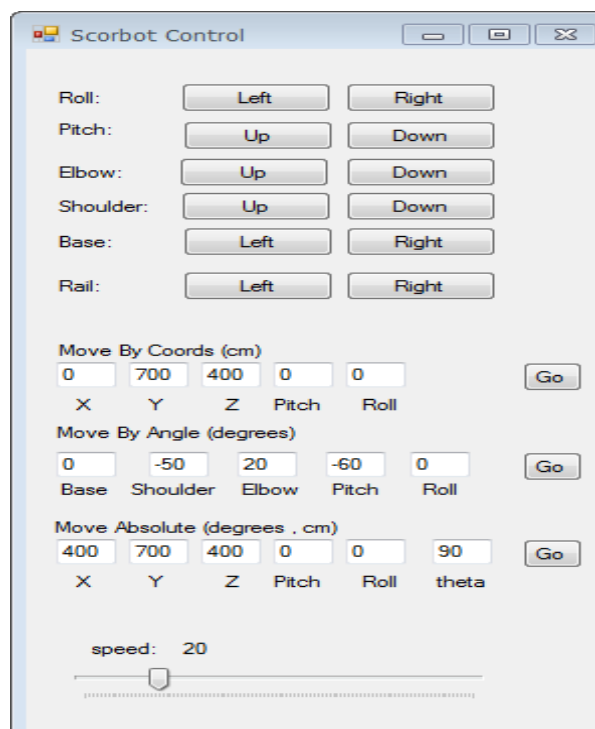
## תקשורת עם הרובוט

לרובוט חמישה צירים ומסילה אותם אפשר להזיז באמצעות של המחובר עליו או ע"י תקשורת סריאלית בשפה מיועדת.



במהלך הפרויקט יצרנו מחלקה המייצגת את הרובוט. המחלקה מייצרת מחרוזות ושולחת אותן דרך יציאת הסריאל. המחלקה מאפשרת שליטה במסילה ובצירי הרובוט וכן להזיז אותו לנקודה במרחב (יפורט בהמשך).

הדרך השנייה לתקשר עם הרובוט היא עם ממשק גרפי (GUI) מיועד המאפשר את הפעולות הנ"ל. להלן תצלום הממשק:

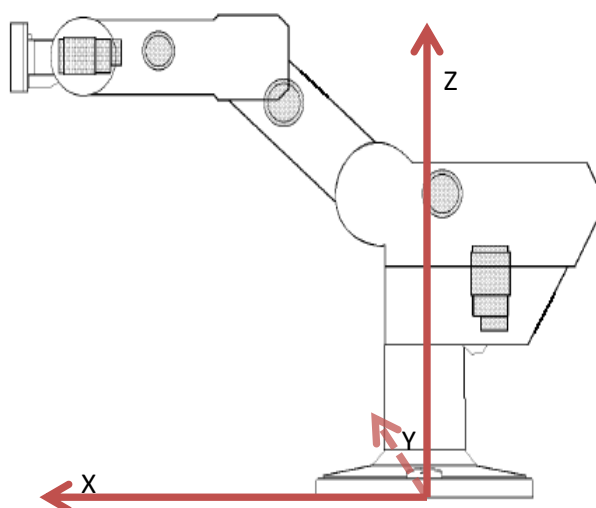


# מעבר מקוארדינטות העולם לזוויות המפרקים של הרובוט (reverse trajectory)

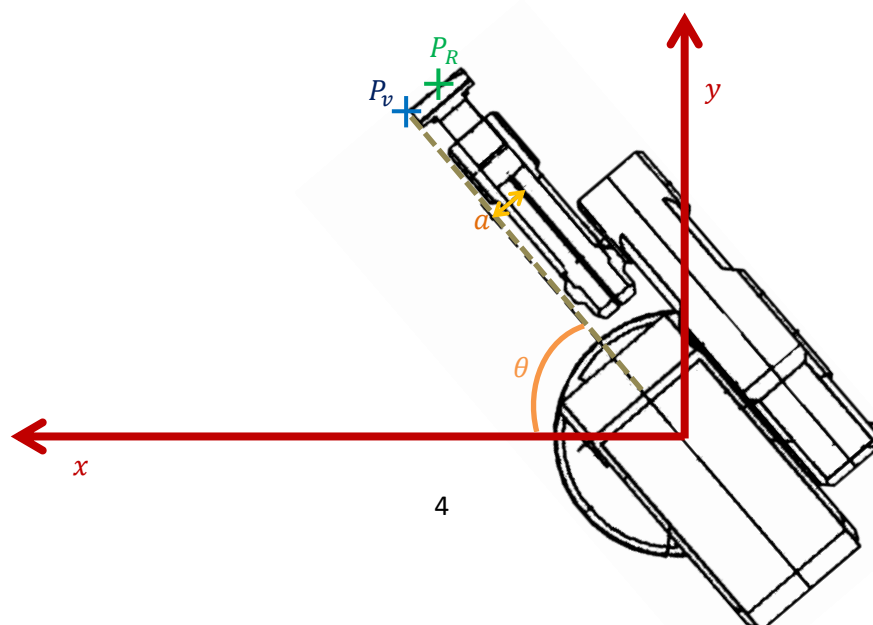
בהינתן נקודה וזווית אנכית (pitch angle), ברצוננו להביא את ראש הרובוט (end effector) לנקודה הנתונה בזווית המתאימה. לצורך כך נחשב את הזוויות של המפרקים.

## חישוב זוויות בשביל נקודה יחסית

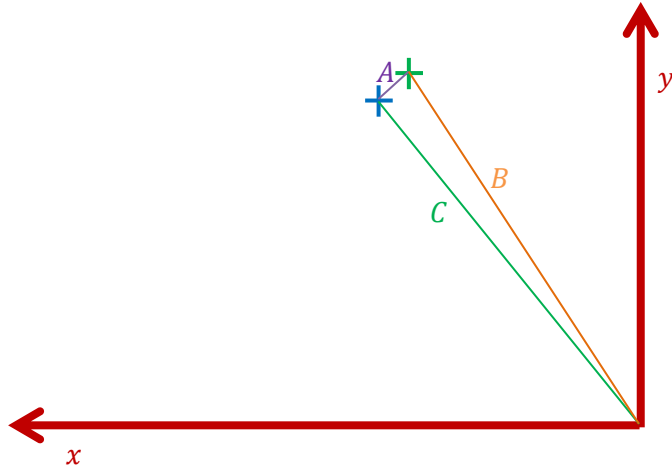
נגדיר נקודה יחסית בתור נקודה הנמצאת במערכת הצירים של הרובוט וביחס לראשית הצירים שלו. מכיוון שמפרקי הרובוט והמסילה הם מערכות נפרדות, נבצע תחילה את המעבר מנקודה יחסית למערכת הצירים הנמצאת במרכז הרובוט.



ראשית, נתבונן על הרובוט ממבט על. עלינו לחשב את הזווית שבה צריך להיות המפרק הראשון על מנת שנוכל להגיע לנקודה.



נרצה לחשב את הזווית  $\theta$  שבה צריך להיות הבסיס. היינו רוצים להתייחס לרובוט כאילו הוא קו ישר, אך ישנו המרחק  $a$  בין מרכז המפרק הסופי לבין המרכז של מפרק הבסיס (מסומן בתרשים). לשם כך נחשב את הנקודה  $P_v$  ונקבע את זווית הבסיס לפי זווית הווקטור של  $P_v$ . נתבונן במשולש שהקדקודים שלו הם  $P_v, P_R$  וראשית הצירים.



ידועים הנתונים הבאים:

$$|A| = a$$

$$|B| = \sqrt{P_R \cdot x^2 + P_R \cdot y^2}$$

$B$  מול  $90^\circ$  = הזווית מול  $B$

מכאן:

$$|C| = \sqrt{|B|^2 - |A|^2}$$

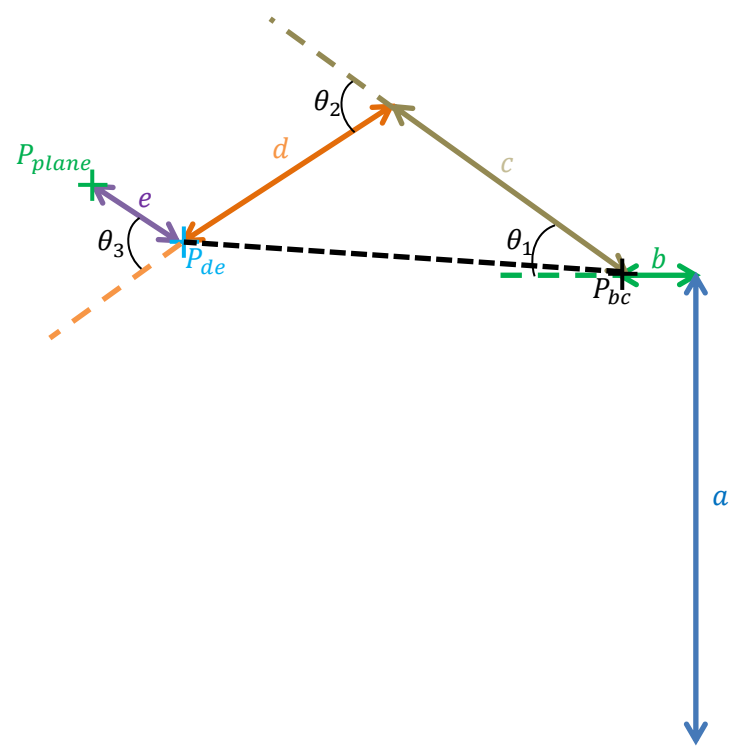
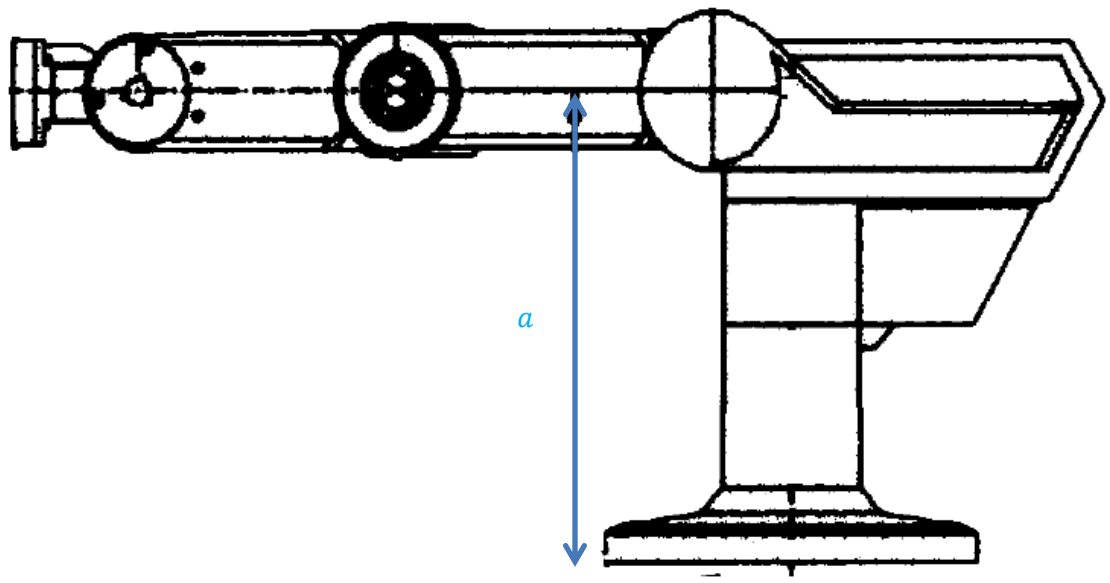
מכאן לפי משפט הקוסינוסים הזווית מול  $A$  (נסמנה ב- $\varphi$ ) נתונה ע"י:

$$\varphi = \cos^{-1} \left( \frac{A^2 - B^2 - C^2}{2BC} \right)$$

נסמן את הזווית של הווקטור  $P_R$  ב- $\gamma$ . מכאן  $\theta = \gamma - \varphi$ . נוכל לחשב את הווקטור  $P_v$  מכך שידוע שאורכו הוא  $|C|$  והזווית שלו היא  $\theta$ .

לאחר שמצאנו את הזווית של הבסיס יש למצוא את זוויות שאר המפרקים. נתבונן במישור הנפרש ע"י ווקטור ציר ה-z והווקטור של  $P_v$ .

נסמן את המפרקים של הרובוט:



נגדיר את הנקודה הקצה של הרובוט (*end effector*) ונסמנה ב-  $P_{plane}$ .

ידוע ש:

$$P_{plane} \cdot x = |P_v|$$

$$P_{plane} \cdot y = P \cdot z$$

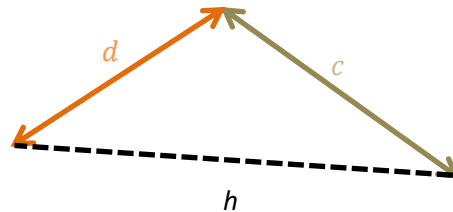
ברצוננו למצוא את זוויות המפרקים.

ניתן למצוא את הווקטור  $e$  ע"י האורך הנתון  $|e|$  והזווית שלו הנתונה כקלט (pitch angle).

לפיכך, ניתן למצוא את נקודת המפגש בין  $d$  ל- $e$  ( $P_{de}$ ) ע"י חיבור הווקטור  $e$  מ- $P_{plane}$ .

כמו כן, ניתן למצוא את נקודת המפגש בין  $c$  ל- $b$  ( $P_{bc}$ ) באמצעות חיבור הווקטורים  $a$  ו- $b$ .

נתבונן במשולש הנוצר ע"י  $c, d$  והצלע בין הנקודות הנ"ל (נסמנה ב- $h$ )



מכיוון וידועות הנקודות  $P_{de}, P_{bc}$ , ניתן לחשב את הווקטור  $h$  שביניהם.

ידועים אורכי הצלעות של המשולש. מכאן בשימוש במשפט הקוסינוסים, כפי שהוצג קודם לכן, ניתן לחשב את הזוויות שמול כל צלע.

ניתן מהמידע הנ"ל למצוא את שלושת הזוויות:

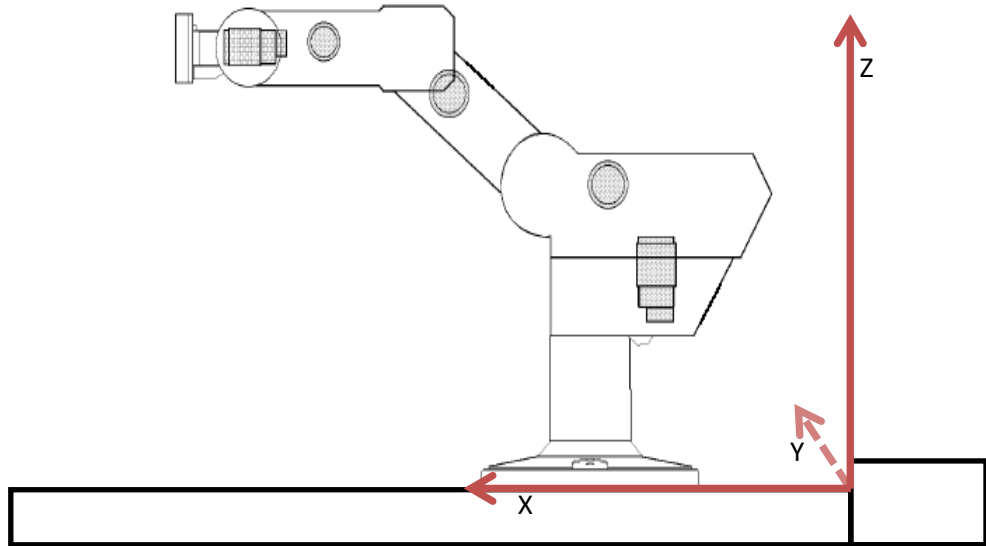
ניתן למצוא את  $\theta_1$  ע"י הוספת הזווית של הווקטור  $h$  לזווית שמול צלע  $d$ .

ניתן למצוא את  $\theta_2$  ע"י חיבור הזווית שמול צלע  $h$  מ- $180^\circ$ .

ניתן למצוא את  $\theta_3$  ע"י הוספת הזווית  $\theta_2 - \theta_1$  לזווית של הווקטור  $e$ .

### חישוב זוויות בשביל נקודה מוחלטת

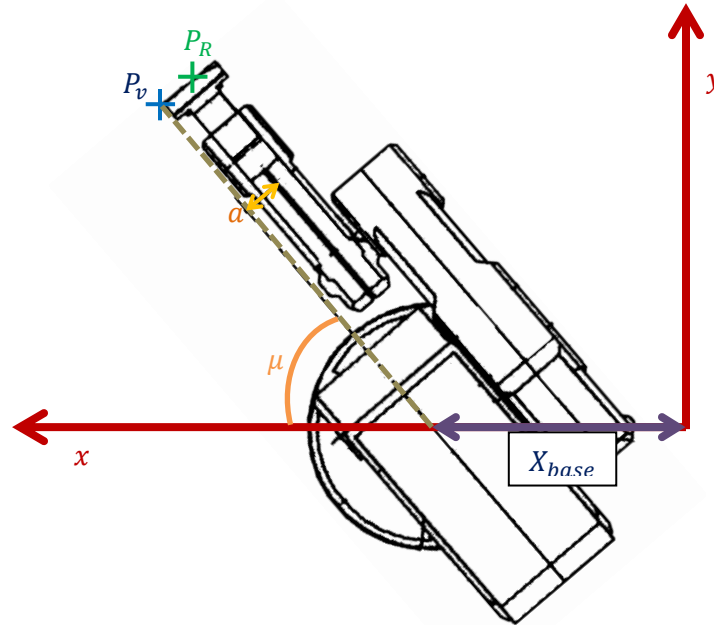
לאחר שהצלחנו להגיע לנקודה יחסית נרצה לחשב את הזוויות הדרושות על מנת להגיע לנקודה המוגדרת ביחס למערכת קואורדינטות המוגדרת על העולם הכוללת למעשה את הנק' במרחב אליהן הרובוט יכול להגיע ע"י הזזת ציריו וכן תנועה לאורך המסילה עליה הוא נמצא.



הפרמטר שנוסף לנו כעת הוא זווית אופקית  $\theta$  שבה נרצה להגיע לנקודה.



נרצה למצוא את  $X_{base}$  (הנקודה אליה רוצים להזיז את הרובוט על המסילה) ואת הנקודה היחסית אליה נרצה להגיד לרובוט להגיע.



ראשית נרצה למצוא את הנקודה  $P_v$ . נסמן ב- $\theta$  את הזווית המשלימה של זווית  $\mu$  המופיעה באיור לעיל.

ניתן להגיע ע"י הוספת ווקטור בעל אורך  $a$  וזווית  $\theta - 90^\circ$  (ווקטור הניצב ל- $P_v$ ) אל הנקודה  $P_R$ . מתקיים:

$$\tan \theta = \frac{P_v \cdot y}{X_{base} - P_v \cdot x} \Rightarrow X_{base} = \frac{P_v \cdot y}{\tan \theta} + P_v \cdot x$$

והנקודה היחסית אליה נגיד לרובוט להגיע היא  $[P_R \cdot x - X_{base}, P_R \cdot y]$ .

# הדמיית ניתוח באמצעות עיבוד תמונה

## הקדמה

הרעיון שעמד מאחורי הפרויקט הוא לבצע הדמיה של ניתוח אדם באמצעות זרוע רובוטית, למשל עבור מטלה של ביצוע ביופסיה בגידול. התהליך שאנו מסמלים כולל צילום של תמונות אבחון, סימון בתמונות האבחון של נקודות עניין שאליה רוצים להגיע בניתוח ("הגידול") ונקודת כניסה שממנה יוחדר כלי הניתוח. בעת ביצוע הניתוח ברצוננו להנחות את זרוע הרובוט כך שתגיע סמוך לנקודת הכניסה בזווית המתאימה לקו המחבר בין נקודת הכניסה ונקודת הביופסיה. על מנת לבצע זאת, נרצה להיות מסוגלים להמיר את נקודת הגידול ונקודת הכניסה שסומנו על תמונות האבחון אל מערכת הצירים של הרובוט.

## רקע

בכדי לבצע את הצילום של אובייקט המטרה עליו אנו מסמלים את הניתוח השתמשנו במצלמה אשר הותאמה וקובעה יחסית למערכת הצירים של הרובוט. בהינתן תמונה של האובייקט (המטופל) השתמשנו במצלמה על מנת למצוא את היחס בין מערכת הקואורדינטות של המטופל לבין זו של הרובוט. כך למעשה היינו בהינתן נקודה במערכת צירים כלשהי היינו מסוגלים לבצע טרנספורמציה ולהעבירה לנקודה המתאימה במערכת הצירים השנייה.

כאשר נתונה מצלמה מסוימת ומערכת קואורדינטות בעולם, יש להתייחס לפרמטרים הבאים:

- $R$  – מטריצת הסיבוב של המצלמה ביחס למערכת הצירים של העולם  
מטריצה זו היא מהצורה הבאה:

$$R_z(\psi) R_y(\theta) R_x(\phi) = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}$$

- $T$  – ווקטור הטרנספורמציה מהמצלמה אל מערכת הצירים של העולם.  
ווקטור זה הוא מהצורה  $T = (x_0, y_0, z_0)^T$  כאשר  $x_0, y_0, z_0$  הם המרחקים של מוקד המצלמה ממערכת העולם בשלושת הצירים.
- $K$  – מטריצת הפרמטרים הפנימיים של המצלמה. מטריצה זו מהצורה הבאה:

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

כאשר:

- $f_x, f_y$  – אורכי המוקד של המצלמה ביחידות של פיקסלים
- $c_x, c_y$  – מיקום המוקד של המצלמה

- $distCoeffs$  – ווקטור פרמטרי העיוות של המצלמה (העיוות העיקרי בגלל אפקט ה-fish eye)

כאשר נתונים פרמטרים אלו ניתן ראשית לתקן את העיוות בתמונה ולאחר מכן, לעבור מנקודה בעולם  $x_w$  לנקודה של המצלמה  $x_c$  בצורה הבאה:

$$x_c = K(R \cdot x_w + T)$$

אנו משתמשים ב- projective plane לכן ניתן להתייחס לנקודה שנקבל אל נקודה דו ממדית באופן הבא:

$$x_c = (x_0, y_0, z_0)^T = (\alpha \cdot x_0^{2D}, \alpha \cdot y_0^{2D}, \alpha)^T$$

הנקודה הדו ממדית המתאימה היא:

$$x_c^{2D} = \left( \frac{x_0}{\alpha}, \frac{y_0}{\alpha} \right)$$

## מטרה

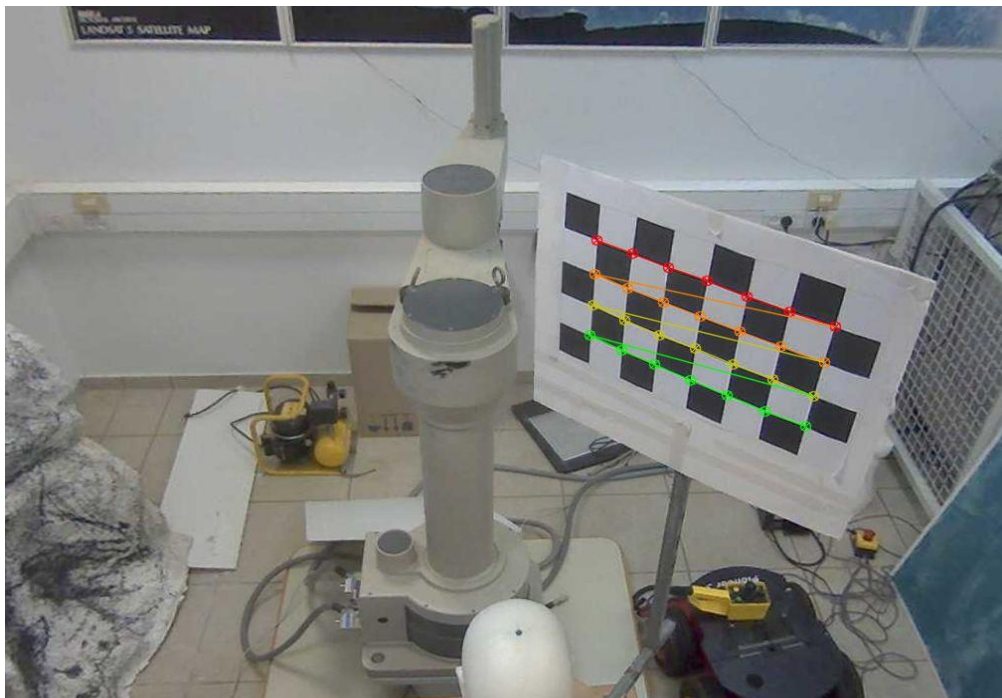
ברצוננו לבנות ממשק המאפשר לבחור שתי נקודות במרחב (נקודת גידול ונקודת כניסה) בתמונות האיבחון. בחירת כל נקודה תבוצע ע"י סימון שתי נקודות דו ממדיות בשתי תמונות, כאשר הראשונה תמונה פרונטאלית והשנייה תמונת פרופיל. לאחר הבחירה, נרצה שהרובוט יזוז לנקודה הראשונה ויצביע לנקודה השנייה.

## מימוש

### קליברציה של המצלמה

ראשית, עלינו למצוא את הפרמטרים הפנימיים של המצלמה ( $K$  ו- $\text{distCoeffs}$ ) לשם כך נעזרנו בפונקציית הספרייה `calibrateCamera` של `opencv` המקבלת בתור קלט נקודות במרחב והנקודות המתאימות בתמונה המתקבלת. (האלגוריתם עובד בצורה יותר מדויקת כתלות במספר הווקטורים של הקלט). על מנת לקבל את הנקודות השתמשנו בפונקציית ספרייה `findChessboardCorners` המקבלת כקלט תמונה ואת הפרמטרים של לוח שחמט (מספר הנקודות הפנימיות לאורך ולרוחב) ומוצאת את הנקודות האלו על התמונה. לשם הנוחות, קבענו את מערכת הצירים בעולם להיות בראשית לוח הצירים של הלוח וכך בכל התמונות יש את אותן נקודות במרחב. השתמשנו בתשע תמונות בתהליך הנ"ל.

דוגמא לפעולת הפונקציה `findChessboardCorners`:



### תיקון עיוות

נשתמש בפונקציית הספרייה `undistort` המקבלת כקלט תמונה, מטריצה פנימית של המצלמה ואת הווקטור `distCoeffs` ומחזירה תמונה ללא עיוות. עבור כל תמונה שנלקחת מהמצלמה אנו משתמשים בפונקציה זו כך שבהמשך לא נצטרך להתחשב ב- `distCoeffs`.

להלן דוגמא:



אחרי



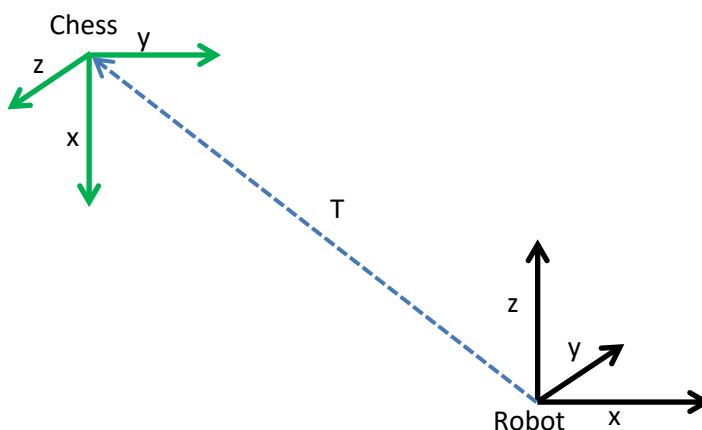
לפני

### מעבר מנקודה במערכת הצירים של הרובוט לזו של המצלמה

נקבע את המצלמה למקום קבוע. כעת נרצה לקבל פרמטרים R ו-T של מעבר מנקודה המערכת הצירים של המצלמה למערכת הצירים של הרובוט. נתאר את תהליך מציאת הפרמטרים הללו.

#### מעבר ממערכת הצירים של הרובוט למערכת הצירים של לוח השחמט

הצבנו לוח שחמט כך שמערכת הצירים שלו ושל הרובוט יהיו מקבילות. לאחר מכן, נציב את ראש הרובוט לנקודת ראשית הצירים של השחמט. נקבל את מערכות הצירים הבאות:



על מנת להעביר נקודה מהמערכת של הרובוט למערכת של הלוח, יש להסתובב  $-90^\circ$  סביב ציר x ולאחר מכן להסתובב  $90^\circ$  סביב ציר z. מכאן מתקבל:

$$R_{Robot}^{Chess} = Q_{90^\circ}(Z) \cdot Q_{-90^\circ}(X) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$

בהינתן שווקטור הטרנספורמציה T הוא מהצורה  $T = (x_0, y_0, z_0)^T$  אז מתקיים  $T_{Robot}^{Chess} = (z_0, x_0, y_0)$ .

#### מעבר ממערכת הצירים של לוח השחמט למערכת הצירים של המצלמה

לשם כך, השתמשנו בפונקציית הספרייה solvePnP של opencv המקבלת כקלט את מטריצת הפרמטרים הפנימיים של המצלמה, נקודות המרחב של לוח השחמט והנקודות בתמונה שעל הלוח (נמצא באופן אוטומטי כפי שהוסבר קודם לכן) ומוציאה כפלט את  $R_{Chess}^{Camera}$ ,  $T_{Chess}^{Camera}$ .

### מעבר מנקודה במערכת הצירים של הרובוט לזו של המצלמה

לאחר שביצענו את החישובים הנ"ל, ידוע כי בהינתן נקודה במערכת הצירים של הרובוט  $x_{Robot}$  ניתן לעבור למערכת הצירים של לוח השחמט באופן הבא:

$$x_{Chess} = R_{Robot}^{Chess} \cdot x_{Robot} + T_{Robot}^{Chess}$$

ובהינתן נקודה  $x_{Chess}$  במערכת הצירים של הלוח ניתן לעבור למערכת הצירים של המצלמה באופן הבא:

$$x_{Camera} = R_{Chess}^{Camera} \cdot x_{Chess} + T_{Chess}^{Camera}$$

מכאן, בהינתן נקודה במערכת של הרובוט  $x_{Robot}$  ניתן לכתוב:

$$x_{Camera} = R_{Chess}^{Camera} \cdot (R_{Robot}^{Chess} \cdot x_{Robot} + T_{Robot}^{Chess}) + T_{Chess}^{Camera} = R_{Chess}^{Camera} \cdot R_{Robot}^{Chess} \cdot x_{Robot} + R_{Chess}^{Camera} \cdot T_{Robot}^{Chess} + T_{Chess}^{Camera}$$

$$R_{Robot}^{Camera} = R_{Chess}^{Camera} \cdot R_{Robot}^{Chess} \quad \text{ו-} \quad T_{Robot}^{Camera} = R_{Chess}^{Camera} \cdot T_{Robot}^{Chess} + T_{Chess}^{Camera}$$

### תהליך בחירת הנקודה

ראשית כל נדאג שהעצם עליו אנחנו רוצים לבחור את הנקודות (בובת ראש במקרה שלנו) יהיה מקובע ללוח שחמט. ניקח שתי תמונות של הבובה, תמונת פרופיל ותמונה פרונטלית, כך שבשתי התמונות לוח השחמט מופיע במלואו.



ראשית נמצא את הפרמטרים החיצוניים של המצלמה -  $R_1, T_1, R_2, T_2$  בכל תמונה. נבחר נקודה דו ממדית  $x_{Camera1}^{2D} = (x_0, y_0)^T$  בתמונה הראשונה ונעביר אותה לנקודה לנקודה  $x_{Camera1} = (x_0, y_0, 1)^T$ . יתקיים עבור כל ערך  $\alpha$ :

$$\alpha \cdot x_{Camera1} = R_1 x_{Chess1} + T_1 \Rightarrow$$

$$x_{Chess1} = R_1^{-1}(\alpha \cdot x_{Camera1} - T_1)$$

כלומר הנקודה הדו ממדית מתורגמת לישר במרחב.  
 לאחר שיש לנו את משוואת הישר, נוכל לבחור שני ערכים  $\alpha_1, \alpha_2$  שרירותיים ולקבל 2 נקודות על הישר  
 $x_{Chess1}(\alpha_1), x_{Chess1}(\alpha_2)$   
 נוכל למצוא את הנקודות הנ"ל בתמונה השנייה לפי:

$$x_{camera2}(\alpha) = R_2 x_{Chess1}(\alpha) + T_2$$

לאחר מכן נצייר את הישר הדו ממדי שיוצרות שתי הנקודות בתמונה השנייה ועל הישר הדו ממדי נבחר נקודה  
 נוספת שגם היא מגדירה ישר במרחב.

$$x_{Chess2} = R_2^{-1}(\alpha \cdot x_{Camera2} - T_2)$$

כעת יש לנו 2 ישרים, שלפי אופן בחירת הנקודות ידוע לנו שהם נחתכים. נמצא את נקודת החיתוך שלהם בצורה  
 הבאה:

- נטיל כל ישר על המישור הדו ממדי כך שנקבל:  
 $l_1 := a_1x + b_1y + c_1 = 0 := (a_1, b_1, c_1)^T$   
 $l_2 := a_2x + b_2y + c_2 = 0 := (a_2, b_2, c_2)^T$
- כאשר משתמשים בprojective plane, נקודה  $p$  נמצאת על ישר  $l$  אם  $l \cdot p = 0$   
 לפי הזהות  $(u \times v) \cdot v = 0$  נסיק כי נקודת החיתוך היא  $l_1 \times l_2$
- כעת מצאנו את היטל נקודת החיתוך על המישור הדו ממדי.  
 נותר רק להציב את הערכים באחד מן הישרים התלת ממדיים ולמצוא את נקודת החיתוך  
 במרחב (ניתן גם לבדוק אם לא התבצעו שגיאות ע"י הצבת הנקודה הדו ממדית בישר  
 השני).

נבצע תהליך זה פעמיים על מנת לבחור שתי נקודות במרחב.

להלן דוגמא:





### הבאת הרובוט לנקודה הרצויה בזוויות הנכונות

הבובה תהיה במרחב העבודה של הרובוט במקום שבו המצלמה קולטת את לוח השחמט.  
 ניקח תמונה ונמצא את הפרמטרים הנוכחיים של לוח השחמט  $R_{Chess}^{Camera}, T_{Chess}^{Camera}$  לפי התמונה.  
 בהינתן נקודה במערכת הקואורדינטות של הלוח במיקום הנוכחי  $x_{Chess}$ , נרצה לעבור לנקודה במערכת  
 הקואורדינטות של הרובוט  $x_{Robot}$ .

מצאנו בשלב קודם את  $K$ ,  $R_{Robot}^{Camera}$  ואת  $T_{Robot}^{Camera}$ . ידוע שמתקיים:

$$x_{Camera} = K(R_{Chess}^{Camera} \cdot x_{Chess} + T_{Chess}^{Camera})$$

וגם

$$x_{Camera} = K(R_{Robot}^{Camera} \cdot x_{Robot} + T_{Robot}^{Camera})$$

מכאן:

$$K(R_{Chess}^{Camera} \cdot x_{Chess} + T_{Chess}^{Camera}) = K(R_{Robot}^{Camera} \cdot x_{Robot} + T_{Robot}^{Camera}) \Rightarrow$$

$$x_{Robot} = R_{Robot}^{Camera^{-1}} (R_{Chess}^{Camera} \cdot x_{Chess} + T_{Chess}^{Camera} - T_{Robot}^{Camera})$$

נעביר את 2 הנקודות שבחרנו למערכת הצירים של הרובוט ונקבל  $x_{Robot1}, x_{Robot2}$ . נבנה ווקטור מ- $x_{Robot1}$  ל- $x_{Robot2}$  וניתן לרובוט פקודה לזוז לנקודה  $x_{Robot1}$  עם הזוויות האנכית והאופקית של הווקטור הנ"ל.

הרובוט מגיע לנקודה הנכונה כפי שהוצג בתמונות מהשלב הקודם.





- Scorbot User's Manual – 2<sup>nd</sup> Edition – Catalog #100066 Rev.B •
- <http://opencv.willowgarage.com/documentation/> - OpenCV 2.0 C Reference •
- Multiple View Geometry in Computer Vision – Second Edition •