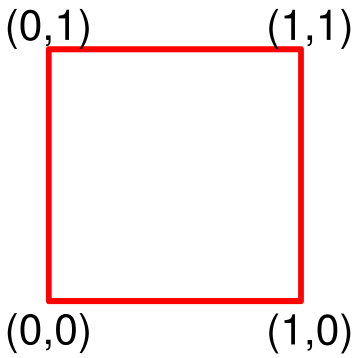
Example Questions - Geometrical Operations

- Warping
- Morphing

Homogeneous Coordinates



Rotate and translate:

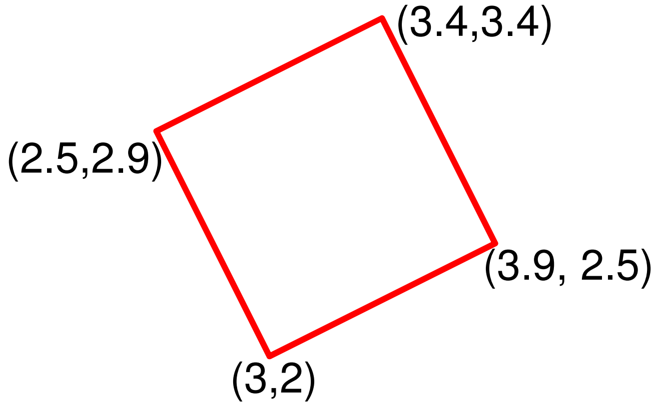
$(tx,ty) = (3,2), \quad \theta = 30^\circ$

$$M = \begin{bmatrix} \cos\theta & -\sin\theta & 3 \\ \sin\theta & \cos\theta & 2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.866 & -0.5 & 3 \\ 0.5 & 0.866 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

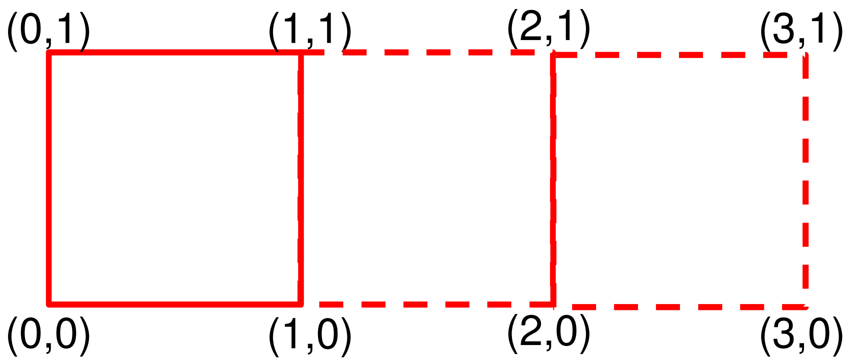
$$\begin{bmatrix} x_i' \\ y_i' \\ w_i \end{bmatrix}_{i=1}^{4} = M \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}_{i=1}^{4} = M \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x_i' \\ y_i' \\ w_i \end{bmatrix}_{i=1}^{4} = \begin{bmatrix} 0.866 & -0.5 & 3 \\ 0.5 & 0.866 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 3.866 & 3.366 & 2.5 \\ 2 & 2.5 & 3.366 & 2.866 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix}_{i=1}^{4} = \begin{bmatrix} 3 & 3.866 & 3.366 & 2.5 \\ 2 & 2.5 & 3.366 & 2.866 \end{bmatrix}$$



Homogeneous Coordinates



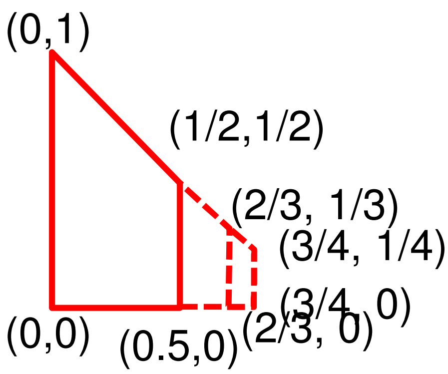
Projective Transformation

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x_i' \\ y_i' \\ w_i \end{bmatrix}_{i=1}^{4} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 0 & 1 \\ 3 & 3 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 0 & 1 \\ 4 & 4 \end{bmatrix}$$

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix}_{i=1}^{4} = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} 2/3 & 2/3 \\ 0 & 1/3 \end{bmatrix} \begin{bmatrix} 3/4 & 3/4 \\ 0 & 1/4 \end{bmatrix}$$



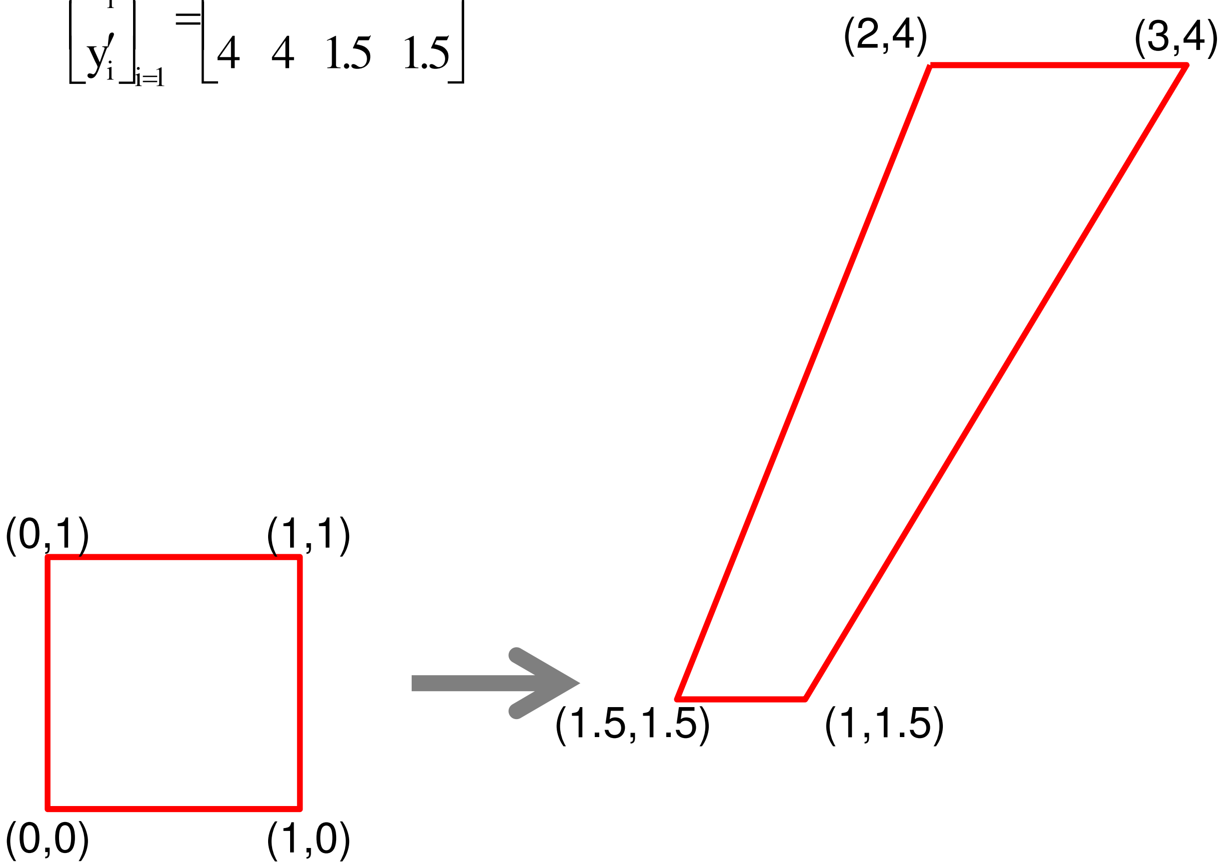
Homogeneous Coordinates

What does this transformation do?

$$M = \begin{bmatrix} 1 & 0 & 2 \\ 0 & -1 & 4 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x_i' \\ y_i' \\ w_i \end{bmatrix}_{i=1}^{4} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & -1 & 4 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 3 & 2 \\ 4 & 4 & 3 & 3 \\ 1 & 1 & 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix}_{i=1}^{4} = \begin{bmatrix} 2 & 3 & 1.5 & 1 \\ 4 & 4 & 1.5 & 1.5 \end{bmatrix}$$



4. נתונות שתי תמונות: תמונה A המכילה עיגול בעל דרגת אפור I_A שמרכזו ב- P_A (וקטור דו-מימדי) ורדיוסו R_A , ותמונה B המכילה עיגול בעל דרגת אפור I_B שמרכזו ב- P_B ורדיוסו R_B .
מעונינים לבצע **morphing** מתמונה A לתמונה B על-ידי יצירת רצף של תמונות I(t), כך שתמונה A היא I(0) ותמונה B היא I(1) (ראה ציור).

א. מצא את הטרנספורמציה הגיאומטרית שמעבירה פיקסל (x,y) בתמונה A למקומו בתמונה I(t)
רמז: מצא מטריצה S(t) שמבצעת scaling ווקטור T(t) שמבצע translation כך ש

$$\begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} = S(t)\begin{pmatrix} x \\ y \end{pmatrix} + T(t)$$

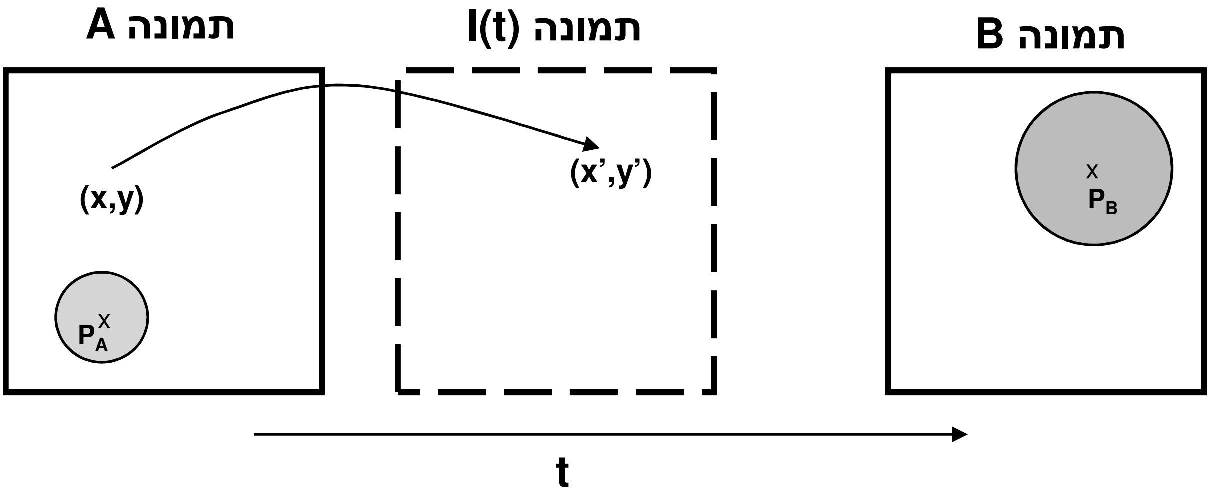
ב. מצא את הטרנספורמציה הגיאומטרית שמעבירה פיקסל (x,y) בתמונה B למקומו בתמונה I(t)

ג. מצא איזה פיקסל מתמונה A מועבר לפיקסל (x',y')בתמונה I(t)

ד. מצא איזה פיקסל מתמונה B מועבר לפיקסל (x',y')בתמונה I(t)

ה. כתוב אלגוריתם שמייצר את תמונה I(t) לכל t נתון בין 0 ל-1. הנח אינטרפולציה של שכן קרוב.

ו. בהנחה שהתמונות מכילות אפסים מחוץ למעגלים, מצא דרך לייעל את האלגוריתם שהצעת בסעיף הקודם.



ב ה צ ל ח ה

4. .א

A point (x_A,y_A) from the source image A, moves to a point (x_B,y_B) in the target image B by the geometric transformation:

$$\begin{pmatrix} x_B \\ y_B \end{pmatrix} = \begin{pmatrix} R_b/R_a & 0 \\ 0 & R_b/R_a \end{pmatrix}\left(\begin{pmatrix} x_A \\ y_A \end{pmatrix} - P_a\right) + P_a + \left(P_b - P_a\right)$$

A point (x_A,y_A) from the source image A, moves to a point (x_t,y_t) in the intermediate image I(t) by the geometric transformation:

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \left[t\begin{pmatrix} R_b/R_a & 0 \\ 0 & R_b/R_a \end{pmatrix} + (1-t)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right]\left(\begin{pmatrix} x_A \\ y_A \end{pmatrix} - P_a\right) + P_a + t\left(P_b - P_a\right)$$

Define:

$$S_{AB}(t) = \left[t\begin{pmatrix} R_b/R_a & 0 \\ 0 & R_b/R_a \end{pmatrix} + (1-t)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right]$$

$$T_{AB}(t) = -S_{AB}(t)P_a + P_a + t\left(P_b - P_a\right)$$

then we have:

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = S_{AB}(t)\begin{pmatrix} x_A \\ y_A \end{pmatrix} + T_{AB}(t)$$

.ב Similarly to .א above we get:

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = S_{BA}(1-t)\begin{pmatrix} x_B \\ y_B \end{pmatrix} + T_{BA}(1-t)$$

where we define:

$$S_{BA}(t) = \left[t\begin{pmatrix} R_a/R_b & 0 \\ 0 & R_a/R_b \end{pmatrix} + (1-t)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right]$$

$$T_{BA}(t) = -S_{BA}(t)P_b + P_b + t\left(P_a - P_b\right)$$

ג.

From א. we extract (x_A,y_A):

$$\begin{pmatrix} x_A \\ y_A \end{pmatrix} = S^{-1}{}_{AB}(t)\left(\begin{pmatrix} x_t \\ y_t \end{pmatrix} - T_{AB}(t)\right)$$

ד.

From ב. we extract (x_B,y_B):

$$\begin{pmatrix} x_B \\ y_B \end{pmatrix} = S^{-1}{}_{BA}(1-t)\left(\begin{pmatrix} x_t \\ y_t \end{pmatrix} - T_{BA}(1-t)\right)$$

ה.

For each pixel (x_t,y_t) in I(t) apply

$$I(t) = (1-t)I_A\big(\mathrm{round}(x_A),\mathrm{round}(y_A)\big) + t\,I_B\big(\mathrm{round}(x_B),\mathrm{round}(y_B)\big)$$

where (x_A,y_A) and (x_B,y_B) are from sections ג and ד above.

ו.

- For each pixel (x_t,y_t) in I(t) find (x_A,y_A) using the equation in ג.
- If $$\left\| \begin{pmatrix} x_A \\ y_A \end{pmatrix} - P_a \right\|^2 \le R_a$$
- $I(x_t,y_t)=t\ I_A + (1-t)\ I_B$ where I_A and I_B are the gray levels of the circle in image A and image B, respectively.
- Otherwise $I(x_t,y_t)=0$

3. נתונות שתי תמונות: תמונה A המכילה קטע ישר בעל דרגת אפור I_A שמרכזו ב- P_A (וקטור דו-מימדי), ארכו L_A, וכוונו מקביל לציר ה-X . תמונה B מכילה קטע ישר בעל דרגת אפור I_B שמרכזו ב- P_B ארכו L_B וכוונו בזוית θ מציר ה-X (ראה תמונה).
מעונינים לבצע **morphing** מתמונה A לתמונה B על-ידי יצירת רצף של תמונות I(t), כך שתמונה A היא I(0) ותמונה B היא I(1) (ראה ציור).

א. מצא את הטרנספורמציה הגיאומטרית שמעבירה פיקסל (x,y) בתמונה A למקומו בתמונה I(t)

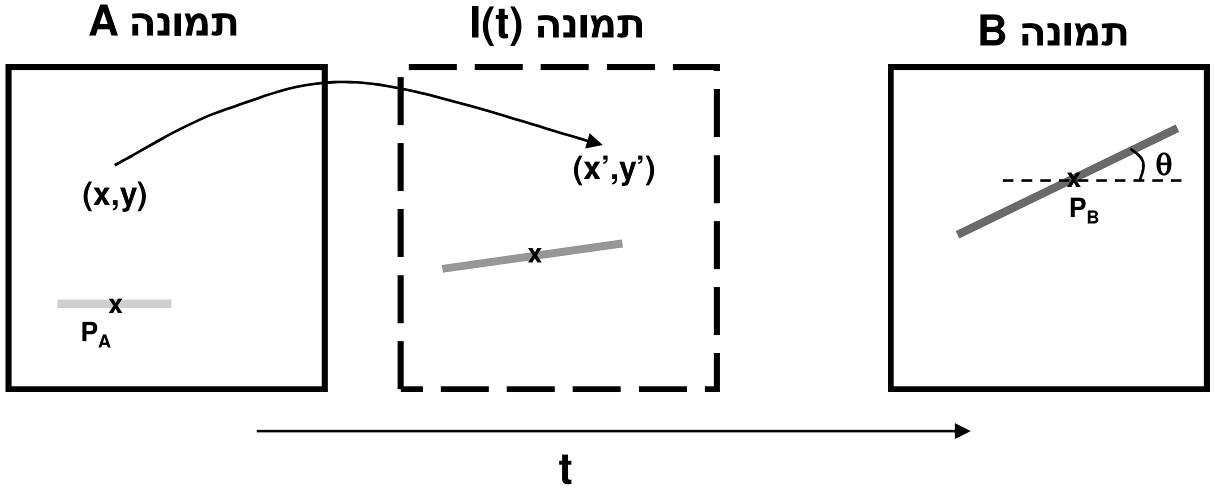
ב. מצא את הטרנספורמציה הגיאומטרית שמעבירה פיקסל (x,y) בתמונה B למקומו בתמונה I(t)

ג. מצא איזה פיקסל מתמונה A מועבר לפיקסל (x',y') בתמונה I(t)

ד. מצא איזה פיקסל מתמונה B מועבר לפיקסל (x',y') בתמונה I(t)

ה. כתוב אלגוריתם שמייצר את תמונה I(t) לכל t נתון בין 0 ל-1. הנח אינטרפולציה בילינארית.

ו. בהנחה שהתמונות מכילות אפסים מחוץ לישר, מצא דרך לייעל את האלגוריתם שהצעת בסעיף הקודם.



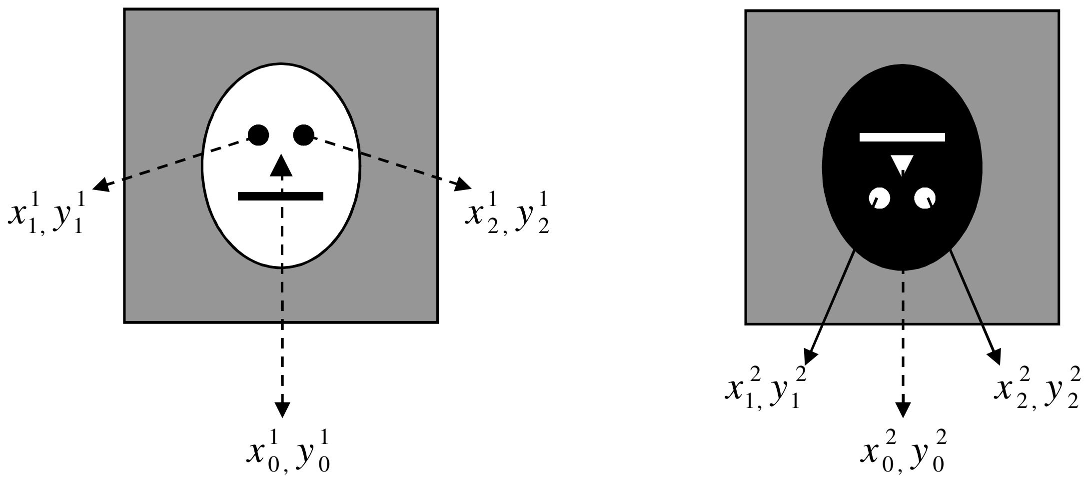
ב. נתונה תמונה של פסיפס **ריבועי** שצולם ממרחק 2 מטר **ו**מגובה 1.5 מטר:

רוצים לשחזר את תמונת הפסיפס במבט על (מעל מרכז הפסיפס).
תאר כיצד ניתן לעשות זאת.
תן תאור מדויק כולל נוסחאות.

שאלה מספר 2

נתונות 2 תמונות I1, I2 :

I1 I2



רוצים ליצור אנימציה (= סדרת תמונות) המתחילה ב- I1 ומסתיימת ב – I2.
זמן ריצת האנימציה מסומנת ב – t (t = 0..1) .
ה – morphing מתבסס על features כך ש- $x^1_{i,} y^1_i$ בתמונה I1 מועתק ל- $x^2_{i,} y^2_i$
בתמונה I2 עבור i = 1..2 .

טרנספורמצית ה- morphing נתונה ע"י מטריצות 2x2 : $\mathbf{S^1_t}$ ו- $\mathbf{S^2_t}$
המופעלת סביב ה"אף" $x^1_{0,} y^1_0$ ו- $x^2_{0,} y^2_0$ בהתאמה
כך שבזמן t הטרנספורמציה המופעלת על I1 מוגדרת:

כל פיקסל (x,y) ב- I1 עובר ל- $\mathbf{S^1_t}[(x_, y) - (x^1_{0,} y^1_0)] + (x^2_{0,} y^2_0)$
ובאופן דומה
כל פיקסל (x,y) ב- I2 עובר ל- $\mathbf{S^2_t}[(x_, y) - (x^2_{0,} y^2_0)] + (x^1_{0,} y^1_0)$

בהכ ניתן להניח כי $(x^2_0,y^2_0) = (x^1_0,y^1_0)$

א. תאר בנוסחה (כפונקציה של $x^2_{i,} y^2_i$ $x^1_{i,} y^1_i$) את $\mathbf{S^1_t}$ ו- $\mathbf{S^2_t}$ עבור t = 0 .
ב. תאר בנוסחה (כפונקציה של $x^2_{i,} y^2_i$ $x^1_{i,} y^1_i$) את $\mathbf{S^1_t}$ ו- $\mathbf{S^2_t}$ עבור t = 1 .

ג. תאר בנוסחה (כפונקציה של $x^2_{i,} y^2_i$ $x^1_{i,} y^1_i$ ו- t) את $\mathbf{S^1_t}$ ו- $\mathbf{S^2_t}$ עבור t כלשהו.
הנח שבזמן t ה-feature $x^1_{i,} y^1_i$ עובר לקואורדינטות שהן t חלקיות הדרך לכוון $x^2_{i,} y^2_i$.

ד. תאר כיצד יראו התמונות בסדרת המורפינג בזמן t=0, 0.25, 0.5, 0.75, 1
(קחו בחשבון גם את ה- cross-dissolve).

שאלה מספר 2

The Matrix $\mathbf{S}_t^1$ transforms pixels of image I1
Transformation is about the nose thus :

$$S_t^1\begin{bmatrix} x - x_0^1 \\ y - y_0^1 \end{bmatrix}$$

Transformation $\mathbf{S}_t^1$ at time t = 0 maps I1 to itself thus

$$S_0^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Transformation $\mathbf{S}_t^1$ at time t =1 maps I1 to I2, thus it maps $x_{i,}^1 y_i^1$ to $x_{i,}^2 y_i^2$ (again remembering that the transforms are about the nose)

$$S_1^1\begin{bmatrix} x_i^1 - x_0^1 \\ y_i^1 - y_0^1 \end{bmatrix} = \begin{bmatrix} x_i^2 - x_0^2 \\ y_i^2 - y_0^2 \end{bmatrix}$$

Solve using pseudo-inverse (as done in HW assignment):

$$S_1^1 = \begin{bmatrix} x_i^2 - x_0^2 \\ y_i^2 - y_0^2 \end{bmatrix} pinv\left(\begin{bmatrix} x_i^1 - x_0^1 \\ y_i^1 - y_0^1 \end{bmatrix}\right) \qquad \text{Eq (1)}$$

Note that **S** is NOT rotation (since rotation would take $x_{1,}^1 y_1^1$ to $x_{2,}^2 y_2^2$).
In fact **S** is reflection about the x-axis (after translation so that the nose is at origin)

Transformation $\mathbf{S}_t^1$ at time t maps $x_{i,}^1 y_i^1$ to a point which is of proportion t on the line from $x_{i,}^1 y_i^1$ to $x_{i,}^2 y_i^2$ i.e to the point:

$$(x_{i,}^1 + t(x_{i,}^2 - x_{i,}^1),\ y_{i,}^1 + t(y_{i,}^2 - y_{i,}^1)) = (1-t)(x_{i,}^1 y_i^1) + t((x_{i,}^2 y_i^2)$$

Thus to get $\mathbf{S}_t^1$ we replace $x_{i,}^2 y_i^2$ of eq (1) above with this new point:

$$S_t^1 = \begin{bmatrix} (1-t)x_i^1 + tx_i^2 - x_0^2 \\ (1-t)y_i^1 + ty_i^2 - y_0^2 \end{bmatrix} pinv\left(\begin{bmatrix} x_i^1 - x_0^1 \\ y_i^1 - y_0^1 \end{bmatrix}\right)$$

Similarly for $\mathbf{S}_{\mathbf{t}}^{\mathbf{2}}$ we have:

$$S_0^2 = \begin{bmatrix} x_i^1 - x_0^1 \\ y_i^1 - y_0^1 \end{bmatrix} pinv\left(\begin{bmatrix} x_i^2 - x_0^2 \\ y_i^2 - y_0^2 \end{bmatrix} \right)$$

$$S_1^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$S_t^2 = \begin{bmatrix} (1-t)x_i^1 + tx_i^2 - x_0^2 \\ (1-t)y_i^1 + ty_i^2 - y_0^2 \end{bmatrix} pinv\left(\begin{bmatrix} x_i^2 - x_0^2 \\ y_i^2 - y_0^2 \end{bmatrix} \right)$$

The images at various times t (assume white = gray value 1, black = gray value0
And background gray is gray value 0.5):