Lab02

NAME ___________________________ NAME ___________________________

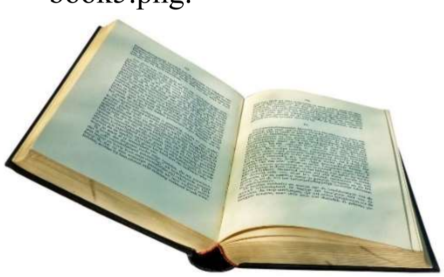
You need to implement either Part 1 (MATLAB-based) or Part 2 (Python and OpenCV), *i.e.*, choose only one of them.

Part 1:

In class, we looked at the hierarchy of 2D coordinate transformations. We saw that each transformation preserves the properties listed in the rows below it (in the table), *e.g.*, similarity preserves not only angles but also parallelism and straight lines. The 2 × 3 matrices are extended with a third $[0^T\ 1]$ row to form a full 3×3 matrix for homogeneous coordinate transformations. For more details, see the textbook (available free online), section 2.1.2.

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[\ \boldsymbol{I} \mid \boldsymbol{t}\ \right]_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\left[\ \boldsymbol{R} \mid \boldsymbol{t}\ \right]_{2\times 3}$	3	lengths	
similarity	$\left[\ s\boldsymbol{R} \mid \boldsymbol{t}\ \right]_{2\times 3}$	4	angles	
affine	$\left[\ \boldsymbol{A}\ \right]_{2\times 3}$	6	parallelism	
projective	$\left[\ \tilde{\boldsymbol{H}}\ \right]_{3\times 3}$	8	straight lines	

- Using the code developed in class, use the appropriate transformation type with the fitgeotrans (Fit Geometric Transformation) function. This function fits a linear geometric transformation of type tform to control point pairs (movingPoints and fixedPoints).
- To create control point, ginput(4) function allows you to identify the coordinates of 4 points. Move your cursor to the desired location and press either a mouse button or a key on the keyboard.
- The target image that we are trying to (fix) transform back is available on Canvas: book5.png.



Choose the appropriate transformation type for each half of the book: one for the left page of the book image and **same or different** type for the right page of the book image.

You may **experiment** with different transformation types and find out which transformation gives a better output.

Turn in one PDF file includes: the code you used and the result images. **Justify** why did you choose a specific transformation type.

Part 2:

Python and Numpy exercises

1. Create a vector A, consisting of the first 5 odd numbers. Create a vector B, consisting of the first 5 even numbers. What is the inner (dot) product of A and B?

2. Find C, the outer product of A and B. What is the sum of all the elements of C? What is the trace of C?

3. Create matrix M with the values shown below. Create M as type "double". What is the product of all elements of M?

$$\begin{matrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{matrix}$$

4. Find the inverse of M. Find the transpose of M. Compute D1, the matrix product of M^{-1} and M^T. Compute D2, the point-by-point product of M^{-1} and M^T. Find $|D1|*|D2|$, where $|.|$ is the determinant operation.

OpenCV exercises

1. Find a nice image of a winter scene online. Write a Python program using OpenCV to read and display the image. Draw a blue snowman on the image, consisting of at least three circles (for the body) and two lines (for the arms).