Image Geometric Transformation

Team - Wednesday

TODO

Practice presentation/demo

Add why this topic is important

Check breakdown from slides

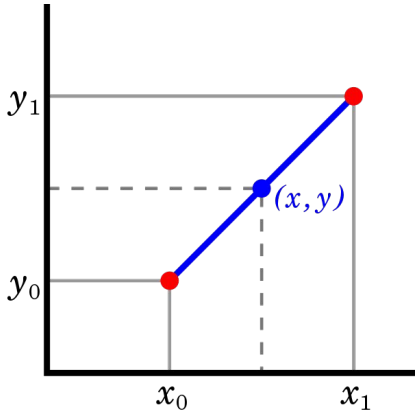
Sub Topics

1. Scaling
2. Translation
3. Rotation
4. Perspective Transform
5. Affine Transformation

Scaling

Interpolation techniques:

- Nearest neighbor
- Linear
- Cubic
- Lanczos4




```
# Nearest Neighbor
Sc = np.zeros(shape=(img.shape[0]*fy, img.shape[1]*fx))
for i in range(Sc.shape[0]):
    for j in range(Sc.shape[1]):
        Sc[i,j] = img[int((i+0.5)/fx), int((j+0.5)/fy)]
```

Translation

```
if ((0 <= (i-u) <= n_rows) and (0 <= (j-v) <= n_rows)):
     Tr[i,j] = I[i-u, j-v]
else:
     Tr[i,j] = 0
```

Rotation

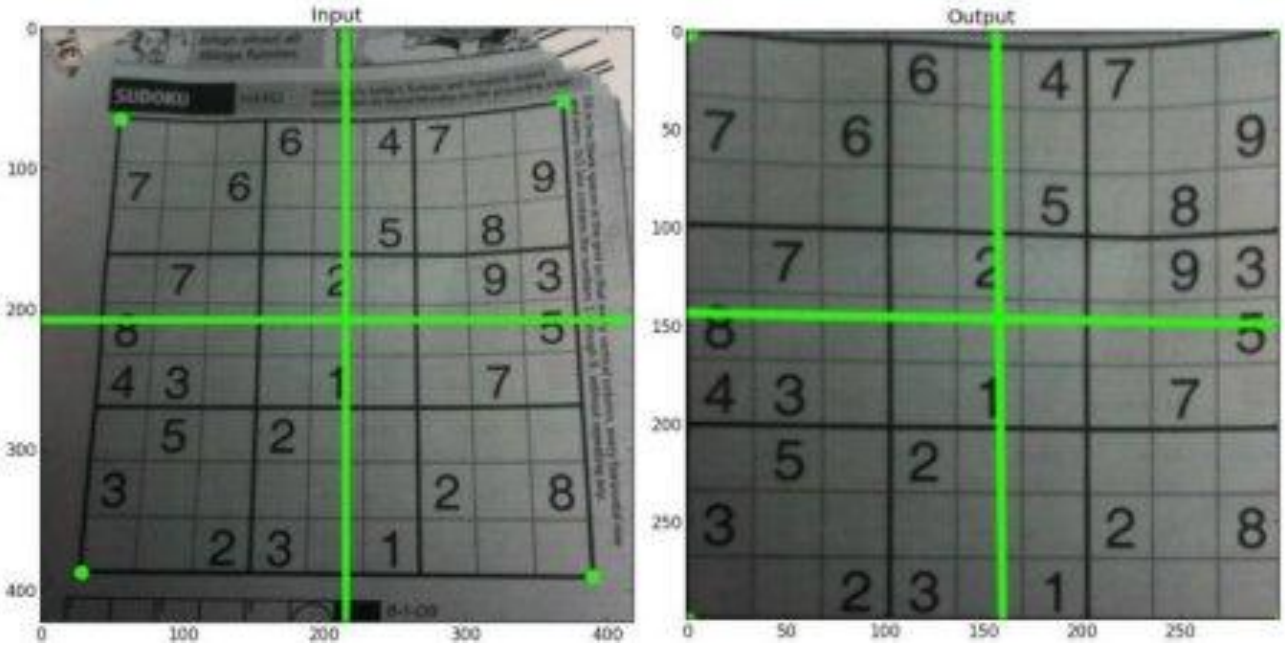
- Basic Rotation Matrix

$$M = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

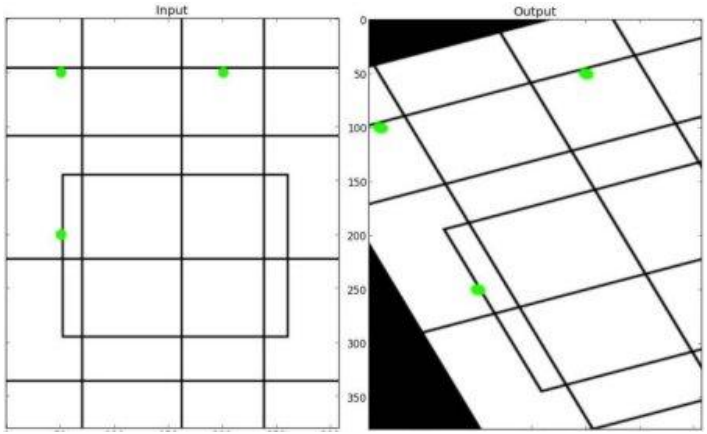
- Scaled rotation about arbitrary center.
  - Scale allows for stretching along a particular component direction

$$\begin{aligned} \alpha &= scale \cdot \cos\theta, \\ \beta &= scale \cdot \sin\theta \end{aligned} \quad \begin{bmatrix} \alpha & \beta & (1-\alpha) \cdot center.x - \beta \cdot center.y \\ -\beta & \alpha & \beta \cdot center.x + (1-\alpha) \cdot center.y \end{bmatrix}$$

Perspective Transform

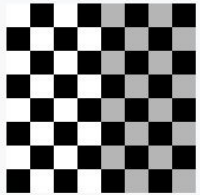
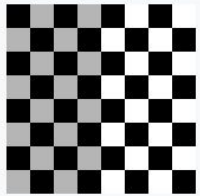
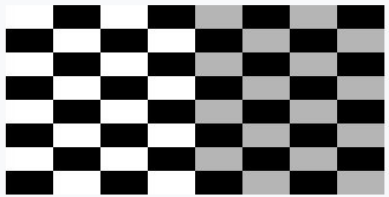


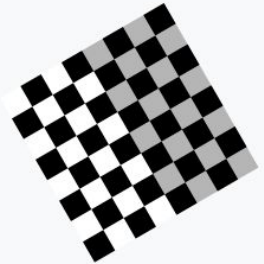
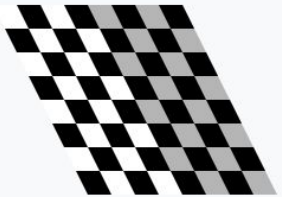
Affine Transform



- Linear Transformation
- Parallel lines remain parallel
- Ratio of distance between points on a line are preserved
- Invertible

$$\begin{bmatrix} i_x & j_x \\ i_y & j_y \end{bmatrix}$$

Identity (transform to original image)	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Reflection	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Scale	$\begin{bmatrix} c_x = 2 & 0 & 0 \\ 0 & c_y = 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	

Rotate	$\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	 where $\theta = \frac{\pi}{6} = 30°$
Shear	$\begin{bmatrix} 1 & c_x = 0.5 & 0 \\ c_y = 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	

Goals

A simple GUI program that performs geometric transformations on a given image.

Implement nonlinear transformations ie swirl or blow up a section

Given a picture from a fisheye lens use warping to fix it

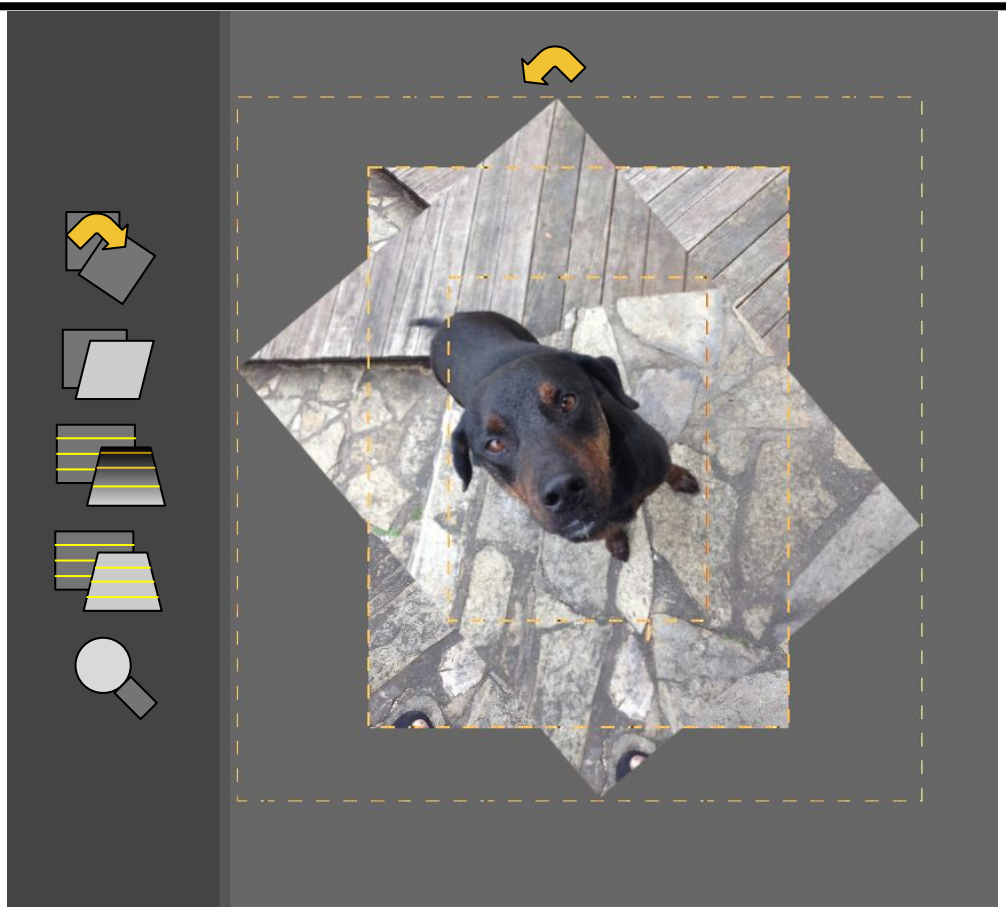
Project Organization

1. Report
2. FrontEnd GUI
3. Library Functions

Library Goals

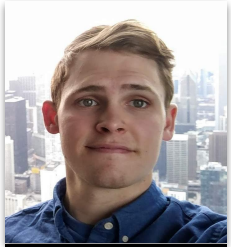
- Implement these CV2 functions

Front end Goals



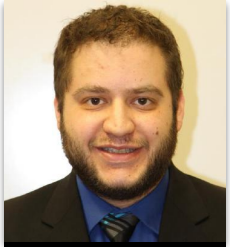
The Team

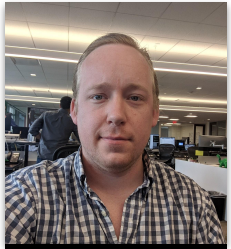

Raymond


Conrad


Christian

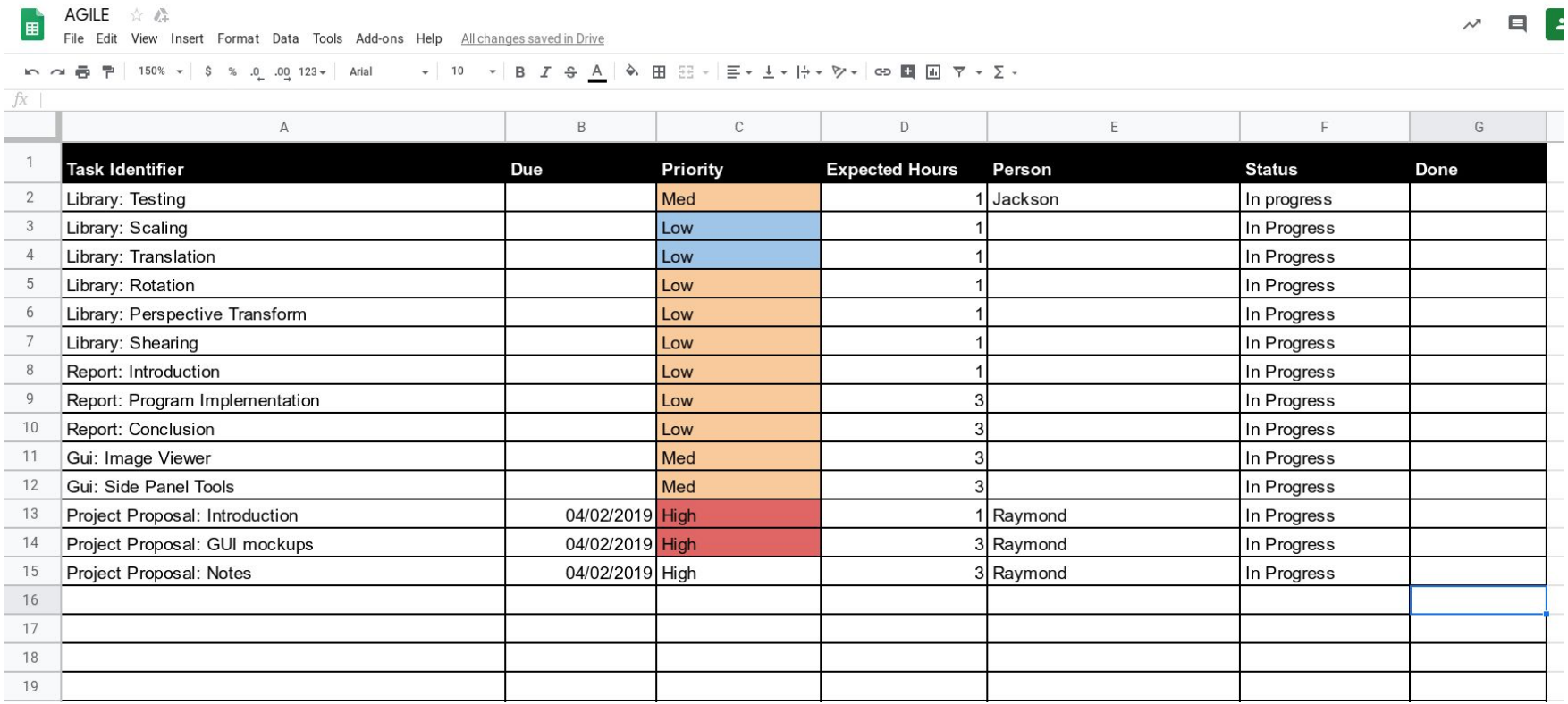

Jackson


Kal


Wynn


Morris

Delegation of Tasks



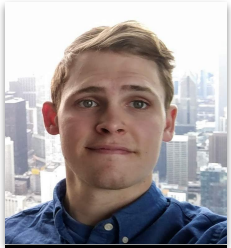
AGILE

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

	A	B	C	D	E	F	G
1	**Task Identifier**	**Due**	**Priority**	**Expected Hours**	**Person**	**Status**	**Done**
2	Library: Testing		Med	1	Jackson	In progress	
3	Library: Scaling		Low	1		In Progress	
4	Library: Translation		Low	1		In Progress	
5	Library: Rotation		Low	1		In Progress	
6	Library: Perspective Transform		Low	1		In Progress	
7	Library: Shearing		Low	1		In Progress	
8	Report: Introduction		Low	1		In Progress	
9	Report: Program Implementation		Low	3		In Progress	
10	Report: Conclusion		Low	3		In Progress	
11	Gui: Image Viewer		Med	3		In Progress	
12	Gui: Side Panel Tools		Med	3		In Progress	
13	Project Proposal: Introduction	04/02/2019	High	1	Raymond	In Progress	
14	Project Proposal: GUI mockups	04/02/2019	High	3	Raymond	In Progress	
15	Project Proposal: Notes	04/02/2019	High	3	Raymond	In Progress	
16							
17							
18							
19							

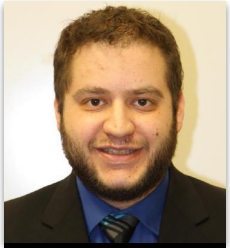
Initial Assignments


GUI


GUI


Library


Testing


Report


Library


Library

Restrictions

Libraries we will use:

- Tkinter
  - for GUI
- Opencv
  - for cv2.imread
- Numpy
  - for matrix operations such as multiply

Libraries we won't use:

- Everything not listed in Libraries we will use