Homework 3 Written Questions

CSCI 1430

Template Instructions

This document is a template with specific answer regions and a fixed number of pages. Given large class sizes and limited TA time, the template helps the course staff to grade efficiently and still focus on the content of your submissions. Please help us in this task:

- Make this document anonymous.
- Questions are in the orange boxes. Provide answers in the green boxes.
- Use the footer to check for correct page alignment.
- **Do NOT remove the answer box.**
- **Do NOT change the size of the answer box.**
- **Extra pages are not permitted unless otherwise specified.**
- **Template edits or page misalignment will lead to a 10 point deduction.**

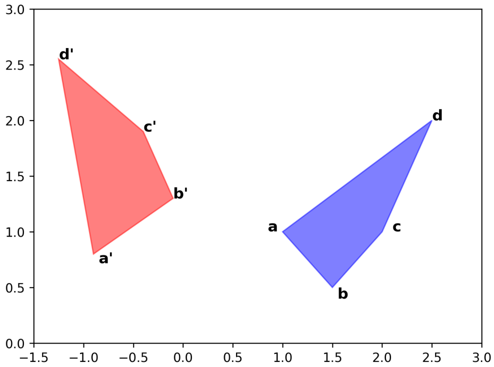
Gradescope Submission

- Compile this document to a PDF and submit it to Gradescope.
- Pages will be automatically assigned to the right questions on Gradescope.

This Homework

- 6 questions **[10 + 8 + 4 + 6 + 3 + 8 = 37 points]**.
- Include code, images, and equations where appropriate.

Q1 — [10 points]
Suppose we have a two quadrilaterals $\mathbf{abcd}$ and $\mathbf{a'b'c'd'}$:



$\mathbf{a} = (1, 1)$ $\mathbf{a'} = (-0.9, 0.8)$
$\mathbf{b} = (1.5, 0.5)$ $\mathbf{b'} = (-0.1, 1.3)$
$\mathbf{c} = (2, 1)$ $\mathbf{c'} = (-0.4, 1.9)$
$\mathbf{d} = (2.5, 2)$ $\mathbf{d'} = (-1.25, 2.55)$

They look like they are related by a rotation and a non-uniform scale transformation. So, let's assume that each point in $\mathbf{abcd}$ could be mapped to its corresponding point in $\mathbf{a'b'c'd'}$ by a 2×2 transformation matrix $\mathbf{M}$, as these can represent non-linear scaling and rotation.

e.g., if $\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$, $\mathbf{p'} = \begin{bmatrix} x' \\ y' \end{bmatrix}$, and $\mathbf{M} = \begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{bmatrix}$,

then $\begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$

Q1.1 — [1 point]

Rewrite the equation $\mathbf{Mp} = \mathbf{p'}$ into a pair of linear equations by expanding the matrix multiplication symbolically.

TODO: Replace each of the '__' below with x, y, x', y', or 0.

$$\begin{cases} x * m_{1,1} + y * m_{1,2} + 0 * m_{2,1} + 0 * m_{2,2} = x' \\ 0 * m_{1,1} + 0 * m_{1,2} + x * m_{2,1} + y * m_{2,2} = y' \end{cases}$$

Q1.2 — [2 points]

We would like to estimate $\mathbf{M}$ using least squares linear regression. For this, we form a system of linear equations, typically written in the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{x}$ contains the parameters of $\mathbf{M}$, matrix $\mathbf{A}$ contains the coordinates of $\mathbf{p}$ and column vector $\mathbf{b}$ contains the corresponding coordinates of $\mathbf{p}'$.

$$\mathbf{Ax} = \mathbf{b} \quad \text{where} \quad \mathbf{A} \times \begin{bmatrix} m_{1,1} \\ m_{1,2} \\ m_{2,1} \\ m_{2,2} \end{bmatrix} = \mathbf{b}$$

As $\mathbf{M}$ has four parameters, we need four equations to *determine* an $\mathbf{M}$. As each point pair provides two equations, that $\mathbf{M}$ will exactly transform those two points $\mathbf{p}$ onto their pairs $\mathbf{p}'$. But, as our quadrilaterals have four points, we could create eight such equations to estimate $\mathbf{M}$—$\mathbf{M}$ is now said to be *overdetermined*. If all point pairs *do not* exactly transform by some $\mathbf{M}$, then using an overdetermined system will find an $\mathbf{M}$ that minimize the distance (or residual error) between all estimated values for $\mathbf{p}'$ and the real values for $\mathbf{p}'$, i.e., that minimize the Euclidean norm (or 2-norm) $||\mathbf{Ax} - \mathbf{b}||_2$.

Note: As systems of linear equations are typically written in the form $\mathbf{Ax} = \mathbf{b}$, we've overloaded the symbol $\mathbf{b}$ here to keep this familiarity. So, be careful—$\mathbf{b}$ is the vector of target x', y' values across all equations, and not the point in the quadrilateral.

Declare $\mathbf{A}$ and $\mathbf{b}$ for all four point pairs.

Replace each '__' below with a 0 or a coordinate value from the quadrilaterals.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1.5 & 0.5 & 0 & 0 \\ 0 & 0 & 1.5 & 0.5 \\ 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 2.5 & 2 & 0 & 0 \\ 0 & 0 & 2.5 & 2 \end{bmatrix} \times \begin{bmatrix} m_{1,1} \\ m_{1,2} \\ m_{2,1} \\ m_{2,2} \end{bmatrix} = \begin{bmatrix} -0.9 \\ 0.8 \\ -0.1 \\ 1.3 \\ -0.4 \\ 1.9 \\ -1.25 \\ 2.55 \end{bmatrix}$$

Only Q1.2 should be on this page

Q1.3 — [3 points]

If we use four equations, then $\mathbf{A}$ is square and can be easily inverted to solve for $\mathbf{x}$ by left multiplying the inverse by both sides:

$$\mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad \text{so} \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}. \tag{1}$$

If we use more than four equations, then $\mathbf{A}$ is non-square. In this situation, we can use the pseudoinverse of $\mathbf{A}$, written as $\mathbf{A}^{+}$.

$$\mathbf{A}^{+}\mathbf{A}\mathbf{x} = \mathbf{A}^{+}\mathbf{b} \quad \text{so} \quad \mathbf{x} = \mathbf{A}^{+}\mathbf{b}. \tag{2}$$

As long as $\mathbf{A}$ has a pseudoinverse, this solution minimizes $||\mathbf{A}\mathbf{x} - \mathbf{b}||_2$.
This is the closed-form least squares solution, where $\mathbf{x} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$ and where $\mathbf{A}^{+} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$.

We can compute the pseudoinverse from the singular value decomposition. In python, `numpy.linalg.lstsq()` will handle this for us, including solving large systems. `lstsq()` takes as input $\mathbf{A}$ and $\mathbf{b}$, and returns a solution for $\mathbf{x}$ along with the residual error. Plug your $\mathbf{A}$,$\mathbf{b}$ values into that function and write the numeric values of the estimated $\mathbf{M}$ matrix below along with the numeric value of the residual error.

Note: We provide a Python script `transformation_viz.py` for you to estimate and visualize the transformation. Within function `estimate_transform()`, declare matrix $\mathbf{A}$ and vector $\mathbf{b}$ and create a function call to `lstsq()` to see how the quadrilaterals match up!

State your $\mathbf{M}$ by replacing each of the '__' below, and state the residual error.

$$\mathbf{M} = \begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{bmatrix} = \begin{bmatrix} 0.332 & -1.0808 \\ 0.876 & 0.1256 \end{bmatrix}$$

Residual error: 0.092

Q1.4 — [4 points]

If the residual is zero (or zero to machine numerical precision), then we can confirm our initial assumption that the transformation is a rotation and a scale. If it is not, then we need a transformation with more degrees of freedom to reduce the residual error.

Determine what kind of transformation it is by forming a system of linear equations and estimating an $\mathbf{M}$ that produces zero residual error (to machine numerical precision).

Use your implementation in `transformation_viz.py` to help. Write out your system's $\mathbf{A}$, $\mathbf{x}$, and $\mathbf{b}$ matrices, state $\mathbf{M}$ and the residual error, and state which kind of transformation it is.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1.5 & 0.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0.5 & 1 \\ 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 1 \\ 2.5 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.5 & 2 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} m_{1,1} \\ m_{1,2} \\ m_{1,3} \\ m_{2,1} \\ m_{2,2} \\ m_{2,3} \end{bmatrix} \mathbf{b} = \begin{bmatrix} -0.9 \\ 0.8 \\ -0.1 \\ 1.3 \\ -0.4 \\ 1.9 \\ -1.25 \\ 2.55 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \end{bmatrix} = \begin{bmatrix} 0.5 & -1.1 & -0.3 \\ 1.1 & 0.1 & -0.4 \end{bmatrix}$$

Residual error: 1.44×10^{-31}

The resulting transformation is a similarity transform.

Q2 — [8 points]
In lecture, we've learned that cameras can be represented by intrinsic and extrinsic matrices. These matrices can be used to calculate the projections of points within a 3D world onto 2D image planes. For this, we use *homogeneous coordinates*. The final 3×4 matrix is known as the *camera matrix*.

Recall that the transformation can be represented by the following expression:

$$\begin{bmatrix} f_x & s & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

where f is the focal length, $\mathbf{R}$ is the rotation matrix, $\mathbf{t}$ is the translation vector, w is some weighing/scaling factor, and (u, v) is the position that the point in the real world (x, y, z) projects to on the 2D plane.

For each following question, you are given the camera specifications and a sample 3D point from the real world. Fill in the camera's intrinsic $\mathbf{K}$ and extrinsic $[\mathbf{Rt}]$ matrices; then, perform the multiplications and perspective division (unhomogenize) to find the 2D coordinate of the projected point.

Q2.1.1 — [1 point]

A camera with a focal length of 1 in both the u and v directions, a translation of 5 along the x-axis, and no skew or rotation.

TODO: Fill in the __ entries.

$$\mathbf{K} \times [\mathbf{Rt}] \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 30 \\ \text{-20} \\ 10 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 35 \\ -20 \\ 10 \end{bmatrix}$$

$$= 10 \times \begin{bmatrix} 3.5 \\ -2 \\ 1 \end{bmatrix}$$

Q2.1.2 — [1 point]

A camera with focal length of 2 in both the u and v directions, a translation of 5 along the x-axis, and no skew or rotation.

$$
\begin{aligned}
&= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 30 \\ \text{-20} \\ 10 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 70 \\ -40 \\ 10 \end{bmatrix} \\
&= 10 \times \begin{bmatrix} 7 \\ -4 \\ 1 \end{bmatrix}
\end{aligned}
$$

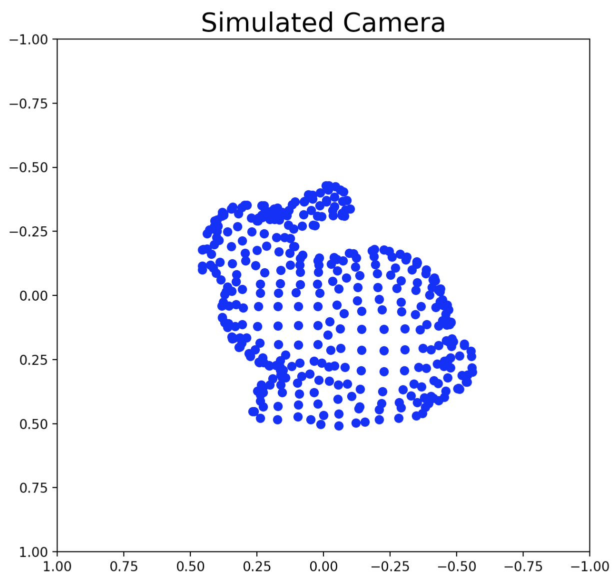
Q2.2 — [1 point]

Compare the two image coordinates you've calculated in parts a and b. What visual effect might we say was induced by this change in intrinsic parameters? **[2–3 sentences]**

The two points are on the same homogeneous plane but the point in part b is scaled further out than the point in part a. This is due to the increase in focal length which results in a zoom-in visual effect of the object.

Q2.3 — [5 points]
We provide stencil code for a camera simulation in `camera_simulation.py`. Given a camera matrix, the simulator visualizes an image that a camera would produce.

Please implement `calculate_camera_matrix()` by calculating the camera matrix using the parameters given in the code. When successful, you will see a bunny rendered as dots (as below). Paste your code for this function and attach a screenshot of the working demo once you finish. Play around with the sliders to see how different parameters affect the projection!



```python
def calculate_camera_matrix(tx, ty, tz, alpha, beta, gamma, fx,
                            fy, skew, u, v):
    ########################
    # TODO: Your calculate_camera_matrix() code here #
    # Hint: Calculate the rotation matrices for the x, y, and z
                                axes separately.
    # Then multiply them to get the rotational part of the
                                extrinsic matrix.
    ########################

    intrinsic_matrix = np.array([[fx, skew, u],
                                 [0, fy, v],
                                 [0, 0, 1]])

    x_rotation = np.array([[1, 0, 0],
                           [0, np.cos(alpha), -np.sin(alpha)],
                           [0, np.sin(alpha), np.cos(alpha)]])
    y_rotation = np.array([[np.cos(beta), 0, np.sin(beta)],
                           [0, 1, 0],
                           [-np.sin(beta), 0, np.cos(beta)]])
    z_rotation = np.array([[np.cos(gamma), -np.sin(gamma), 0],
                           [np.sin(gamma), np.cos(gamma), 0],
                           [0, 0, 1]])
    R = x_rotation @ y_rotation @ z_rotation

    t = np.array([[tx], [ty], [tz]])

    extrinsic_matrix = np.hstack((R, t))

    camera_matrix = intrinsic_matrix @ extrinsic_matrix


    return camera_matrix, intrinsic_matrix, extrinsic_matrix
```

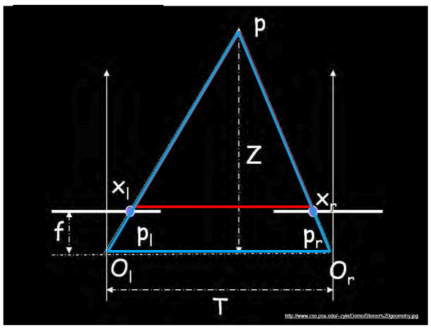
```
###############################################
# YOU MAY USE THIS ADDITIONAL PAGE

# WARNING: IF YOU DON'T END UP USING THIS PAGE
# KEEP THESE COMMENTS TO MAINTAIN PAGE ALIGNMENT
###############################################
```

Q3 — [4 points]
Let's consider two-view geometry.

Q3.1 — [2 points]

Given a stereo pair of calibrated cameras, briefly describe depth estimation by triangulation from disparity. Describe the inputs and outputs of the process. You may wish to use a diagram. **[3–4 sentences]**



Depth estimation by triangulation from disparity is a method to reconstruct 2D image points in 3D space by relating the two images produced by the stereo pair using similar triangles. The method uses the inverse relationship between disparity ($x_r - x_l$) and the focal lengths (f) of the cameras to determine the depth of image point (Z) in the 3D reconstruction. The inputs are the location of the camera centers (O_l and O_r), the distance between the camera centers (T), the 2D coordinates x_l and x_r of p_l and p_r which are the projections of point p onto the camera images, and the focal length of the cameras (f). The output is the 3D coordinates of the image point (p).

Q3.2 — [2 points]

We wish to estimate depth in real-world units (e.g., meters) by triangulation. This is sometimes called ‘absolute depth.’ What parameters or measurements of my stereo pair camera system do I need to know to estimate absolute depth? Provide units. **[3–4 sentences]**

We would need to know the intrinsic attributes of both cameras and the distance between the two cameras. The focal length (intrinsic) and distance between the camera centers would be measured in meters or millimeters. The disparity between the corresponding points would be measured in pixels.

Q4 — [6 points]
Given the algorithms that we've learned in computer vision, we know that whether we can estimate the essential matrix, the fundamental matrix, or both depends on the setup of the cameras and images. Suppose we have three captures of an object of unknown size, without any other metadata (e.g., no Exif metadata):

(i) A video of a camera circling the object;
(ii) A stereo pair of calibrated cameras that captured two images of the object; and
(iii) Two images on the Web captured at different times, with different cameras, and from different locations, with overlap in their views (e.g., Colosseum in Rome).

Q4.1 — [3 points]

For each of the above setups, what can we recover?
Provide brief explanations to support your choices. **[1–2 sentences]**

(i) Setup 1

- ☐ Essential Matrix
- ☑ Fundamental Matrix
- ☐ Both

From the video, we can pull still images such that we can correspond points on the images. But, since we don't know the intrinsic attributes of the camera, we cannot estimate the essential matrix.

(ii) Setup 2

- ☐ Essential Matrix
- ☐ Fundamental Matrix
- ☑ Both

Since the two cameras are calibrated, we know the intrinsic attributes and can estimate the essential matrix. We can also estimate the fundamental matrix since we know the essential matrix and the intrinsic matrix.

(iii) Setup 3

- ☐ Essential Matrix
- ☑ Fundamental Matrix
- ☐ Both

Since we did not calibrate the cameras, to be able to relate corresponding points between the pairs of images, we need to estimate the intrinsic attributes of the two cameras as well.

Q4.2 — [3 points]

State one practical advantage and one practical disadvantage of using each setup for depth reconstruction. **[2–3 sentences]**

(i) Setup 1

A key advantage is that we will have multiple viewpoints of the object giving us more information for a better depth reconstruction. However, a key disadvantage is that it is hard retrieve still images that are clear and can be used for correspondence as is not a trivial task.

(ii) Setup 2

An advantage is that we know how the two images are related to each other since the two cameras are calibrated. However, since are only getting two viewpoints of the object, the camera might miss clear indicators of depth due to occlusions or being too close to or far from the object.

(iii) Setup 3

An advantage is that we have more information about the object because of the key differences of the images. However, a key disadvantage is correspondence matching between the two images. The differences in the images could lead to incorrect or incomplete features matches leading to an incorrect depth reconstruction.

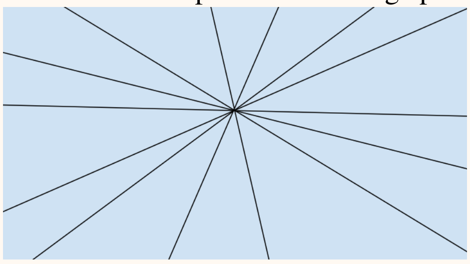
Q5 — [3 points]
In two-view camera geometry, what do the following epipolar lines say about the relative positions of the cameras? **[1–2 sentences each]**

Please draw or describe the relative positions of the two camera planes.

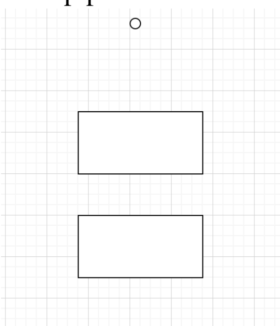
Note: The Spring '22 course staff created an interactive demo to explore the different scenarios. Move the cameras around and look at how the epipolar lines change to gain a better feel for epipolar geometry.

Q5.1 — [1 point]

Radiate out of a point on the image plane.

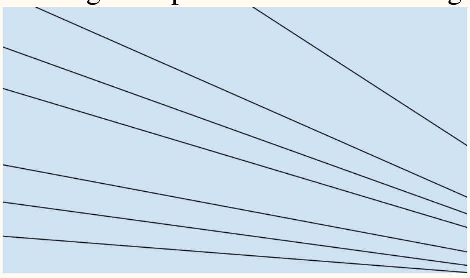


The epipolar lines indicate that the two cameras are placed one in front of another.

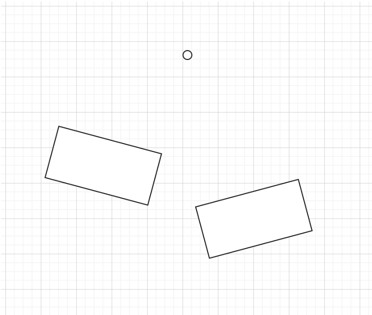


Q5.2 — [1 point]

Converge to a point outside of the image plane.

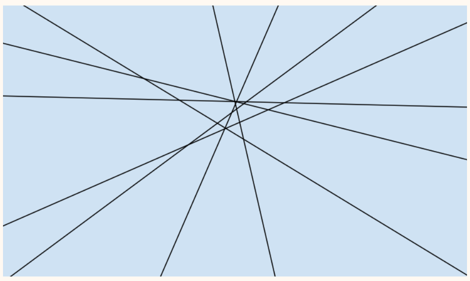


The two cameras are slightly angled to face each other, but the right camera is places further away from the object than where the left camera is placed.



Q5.3 — [1 point]

Notice the misalignment of the epipolar lines in the image below? What went wrong in the calculation of the fundamental matrix and how can we fix it?
Hint: Check slides from the lecture on stereo geometry.



The two cameras are placed one in front of another, but the when estimating the fundamental matrix, it was not enforced that the matrix has a rank of 2 (i.e $det(F) \neq 0$)

Q6 — [4 points]

Cameras are used in surveillance systems. One argument in favor of surveillance systems is to deter and solve crime to improve safety. Another is that if you're not doing anything wrong, you don't have anything to worry about. One argument against surveillance systems is that they compromise people's privacy even when no wrongdoing is taking place. Another is that they increase stress and anxiety.

Computer vision allows the *automation* of surveillance. For instance, it lets us find the mathematical relationship between multiple cameras to track objects and people in 3D spaces, or it can reduce the burden upon a human operator who need only respond to detected events rather than actively monitor many cameras. Such functionality makes it easier to scale a surveillance operation.

On Brown's campus, the number of surveillance cameras has increased as technology has advanced: from 60 in 2000 (Brown Daily Herald 2008/01) to 800 by 2020 (BDH 2020/02), including temporarily in residence halls (BDH 2021/07). The City of Providence is investigating installing noise cameras (BDH 2024/03), potentially on Thayer Street (BDH opinion 2024/04).

Suppose Brown both did and did not use computer vision automation. How comfortable are you with Brown's surveillance apparatus in each case? In what circumstances do you believe that the potential benefits of surveillance *automation* outweigh the potential concerns, and why? **[8–10 sentences]**

Personally, I am more comfortable with the usage of computer vision for surveillance than having a person surveilling. Computers don't inherently misuse the data that is gathered. All they do is perform the task that they are programmed to do and it is the people instructing the software that leads to the compromise of people's privacy. In the case where computer vision is used for surveillance, it is just a number of mathematical computations to determine if a particular scene matches a known wrongdoing scene. If the system doesn't detect any wrongdoing, it ”forgets” that data and moves on, and, if trained correctly, it will do the surveilling in an unbiased manner. If a human were to do the surveilling, they would have their own biases and stereotypes they would places on everyone they are seeing possibly leading to an incorrect view of what is happening. I would be more uncomfortable knowing that someone is watching my every move and possibly using it against me. To me, surveillance automation outweighs the potential concerns in almost every scenario. Like mentioned before, the nefarious activity leading to the comprising of privacy is due to the people behind the automation system and not the system itself and if the automation doesn't exist, the people behind the system would be doing the same thing. Automation also allows for more precise detection as there is always human error.

Q6.1 — [4 points]

Unmanned aerial vehicles—sometimes called drones—often carry cameras. Their cameras can be used for navigation via manually remote control, or for use within computer vision strategies like camera pose estimation and depth estimation to enable assisted or autonomous flying in complex environments.

For your CSCI 1430 final project, you are developing a drone for life-saving organ delivery. You create a successful computer vision algorithm that allows your drone to navigate autonomously. You are approached by several organizations that want to pay you for access to your project, but you are also considering open sourcing your algorithm with a permissive software license.

Please list three organizations that might be interested in acquiring your project for their own purposes. If each of these organizations used your project, who could benefit and how? Who could be harmed and how? Would an open-source license affect this? **[6-9 sentences]**

One organization that would be interested is UPS or similar delivery companies where both the customers and the organization would benefit from project as goods can be delivered faster and more efficiently. However, this would mean having drones fly around in residential communities where neighbors and other community members might not be fine with having something with surveillance capabilities going around their community for privacy reasons. An open-source license could potentially allow community members to add the safe guards that make them more comfortable with the technology, but adversaries would also have access to it and can use it in a negative way. Another organization that benefits from this technology is healthcare providers where, like described in the example, an autonomous drone can allow for organ and other medical supplies to be delivered efficiently. However, the patients receiving care could be harmed if the drones' software is not regulated. Having an open-source license would only make this worse as malicious actors could gain access to the drones and tamper with it. A third organization that benefits from the autonomous technology is government organizations like FEMA where the drones would help with the relieve efforts faster than getting boots on ground. However, this could negatively affect communities if the government uses the drones for surveillance efforts. An open-source license would only make this worse by allowing access to nefarious actors.

Feedback? (Optional)

We appreciate your feedback on how to improve the course. You can provide anonymous feedback through this form, that can be accessed using your Brown account (your identity will not be collected). If you have urgent non-anonymous comments/questions, please email the instructor.