# CSE 559A Computer Vision

**FL2020:** Tue/Th 11:30-12:50 @ Wrighton 300 / [Zoom](#)

This course introduces the fundamentals of designing computer vision systems—that can "look at" images and videos and reason about the physical objects and scenes they represent. We will learn about methods for image restoration and enhancement for estimating color, shape, geometry, and motion from images; and for image segmentation, recognition, and classification. The focus of the course will be on the mathematical tools and intuition underlying these methods: models for the physics and geometry of image formation, and statistical and machine learning-based techniques for inference.

* Caveat: The schedule is tentative, and may change over the course of the semester.

Problem sets, proposals, and reports are due at 11:59 PM (Central) on the dates indicated.

The first five Classes before the drop deadline will be purely online over Zoom. After the drop deadline, you will have the option of attending in-person or through Zoom. Those attending asynchronously can access cloud recordings from the [Zoom tab in Canvas](#).

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep 15 (Online only)</td>
<td>PS0 Out</td>
<td>Sep 17 (Online only)</td>
<td>Image Formation II. Per-pixel Image Operations.</td>
</tr>
<tr>
<td></td>
<td>Introduction. Cameras &amp; Image Formation I.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sep 22 (Online only)</td>
<td>PS0 Due PS1 Out</td>
<td>Sep 24 (Online only)</td>
<td></td>
</tr>
<tr>
<td>Sep 29 (Online only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oct 6</td>
<td>Restoration &amp; Optimization II.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oct 8</td>
<td>Color &amp; Illumination. Material &amp; Shading: Lambertian, specular, BRDFs.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oct 13</td>
<td>PS1 Due PS2 Out</td>
<td>Oct 15</td>
<td></td>
</tr>
<tr>
<td>Oct 20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Policies

**Grade:** The grade will be based on five problem sets (15% each for a total of 75%), and a final project (25%). There will be no exams.

**Late Policy:** All problem sets are due by 11:59 pm of the due date (although we don't suggest waiting till then to submit). Any problem set that is turned in more than 3 days late will simply not be graded. Problem sets turned in 1-3 days late will be penalized according to the final policy.

You get three free late days total for the course. Beyond that, you will lose 25% for each day a submission is late (this is quantized to days: anything that is submitted between 1 minute and 24 hours after the due date will be penalized 25%). Late penalties will not be reflected in the initial grade for each set, but will be applied at the end of the term after factoring in the late days (in a way that maximizes your points).

This policy only applies to problem sets. Project proposals and reports MUST be submitted on time (unless you have prior permission from the instructor, which will rarely be given).
Collaboration and Academic Honesty: We have zero tolerance for academic integrity violations. We actively use automated tools to detect copying. All suspected violations will be reported to the appropriate university office based on the student’s department. Students whose violations are confirmed will receive an F in the course: irrespective of how many points the affected problem was worth. There will also be likely additional serious sanctions from the university. It is therefore extremely important that you read through the collaboration policy below very carefully. If in doubt, ask us if something is allowed.

All problem sets, and the final project, are expected to be completed individually. All code and written work you submit must be your own. Discussion about course topics with your classmates is encouraged (in person, and on piazza), but remember:

1. Do not look for answers to the problem set questions online!
2. Do not share your code, even individual lines of code, for any of the problem sets with anyone (to help them or to seek help).
3. Do not look at code for any problem sets provided written by another person: by current/past students of the course, found online, etc.
4. Do not use external libraries or code shared online in any of the problem sets. It is fine to look at general python/numpy/scipy manuals, but not for specific solutions to the questions. When in doubt, ask the course staff!
5. Do not carry away any written notes, pictures of a whiteboard, or examples of code away from a discussion session. And do something else for an hour before going back to the problem set. Make sure you aren't solving the problem set from memory.
6. In every problem set you turn in, acknowledge any sources of assistance, as well as names of classmates you had a discussion with. But these sources and discussions must nevertheless follow this policy. Acknowledgments don't excuse violations.
7. You may use external resources (including code) ONLY FOR THE FINAL PROJECT when appropriate, provided these sources are also properly acknowledged. When writing your final project report, do not copy sections directly from the original papers on the method—explain these in your own words. If you are repeating some text verbatim, put it in quotes and add a citation. If you are repeating equations directly from the original paper, also make this clear and include a citation.

Follow both the letter and spirit of this policy. The problem sets account for most of your grade, and it is important we be able to evaluate how well you, personally, have understood the course material. Discussions with classmates should be to gain a better understanding of the course material in general, not the specific problems in the problem sets. If you are stuck on a problem, ask for help from the course staff instead of from your classmates (or online). If a classmate asks for help and you can’t be sure you can help them without "revealing" a crucial part of the answer, ask them to contact us.

Start early on problem sets, and if you need help, ask us. And remember, we routinely give generous partial credit for incomplete, buggy, or partially-correct code and solutions, as long as it is your own work!

Technical Requirements

Students needing financial assistance to acquire the necessary hardware below should contact the Office of Student Success.

Homeworks: To complete problem sets and the final project, you will need access to a laptop or computer. All required software will be open source. You will also need internet access to retrieve and submit homeworks.

Lectures: You will need a computer with a stable internet connection to synchronously attend lectures. Note that even if you are planning to attend in-person, the first few classes before the drop deadline will be purely online. You do not need a webcam or microphone to attend lectures—we will mostly rely on text-based chat for students to ask questions, answer questions posed in class, etc. If you plan to attend in person, it would be useful (but not necessary) to have a laptop or smartphone, to be able to answer questions posed during class. If you are unable to attend synchronously (e.g., due to timezone issues), cloud recordings of class sessions as well as slides will be available for download in Canvas.

Asking Questions & Office Hours: We highly recommend using Piazza to ask questions outside of class. The instructor will monitor and respond to questions in Piazza frequently. Please avoid e-mailing the instructor, instead post a private question on Piazza. We will also have office hours held over Zoom. To attend these, you will need access to a computer with at least a microphone (a webcam is useful, but not required).

Zoom: Links to all Zoom sessions and recordings will be available through Canvas. These sessions and recordings are intended for use of current students only and neither meeting invites nor recorded sessions should be shared or posted elsewhere for any purpose.

Overseas Technology Access: If you will be based overseas, there may be technology access considerations. Contact the instructor if you have persistent issues accessing the course online resources.
Problem Sets

Problem sets will be distributed and collected through GIT repositories. Please install git on your machine, if it's not already there. You will access the homework repositories using SSH using public key authentication. If you are not familiar with this, look at this tutorial. All enrolled and wait-listed students will receive a link by e-mail to provide us with their public key.

Once we have received your public key, you will be able to access each homework, once it's been posted, by "cloning" a repository as:

```
$ git clone submit@cse559.ayanc.org:wustl.key/psetN
```

Here, replace `wustl.key` with the username part of your WUSTL key. This is NOT your numeric student ID—instead, it is often (but not always) the part before the `@wustl.edu` of your e-mail address. When we send you an e-mail to submit your public key, that e-mail will also make a note of your `wustl.key` for your reference. Also in the clone command above, `psetN` will be name of the problem set (e.g., `pset0`, `pset1`, etc.). Once you do a clone, you will see the repository inside a new sub-directory `psetN` where you ran the clone command.

The repository will contain the problem set in a file called `pset.pdf`, starter code and data in the `code/` directory, as well as a TeX template for the solutions (`solution.tex`) and the corresponding PDF (`solution.pdf`). Once you have completed your homework, you should add, commit, and push your solutions to the repository server. Make sure you add the write-up `solution.pdf` as well as the updated `.py` files in the `code/` directory. You should not add the generated output image files to the repository (these should be included in the write-up file).

Note that just committing the files is not enough, because that just updates the local repository on your machine. You must push the files for your submission to be complete. After the repository receives your push, it will create an empty commit on top with comments detailing the timestamp and the list of files receive. So, you need to do a pull immediately after, and use `git log` to verify your submission. The whole process will look something like this (run this within the `psetN` sub-directory created by clone command):

```
$ git add solution.pdf code/*
$ git commit -m "submission"
$ git push
$ git pull
$ git log
```

Once we have graded your problem set, the grades will be posted on Canvas, and a file with your grade + comments will be added to the repository. You can do a `git pull` to retrieve these.

The coding portion of the assignments will require the use of the Anaconda distribution of Python + NumPy, SciPy. Please download the Python 3.6+ version of the [latest Anaconda distribution](https://www.anaconda.com/products/distribution). The write-ups for all homeworks must be done in LaTeX. Please see the LaTeX portion of the resources section if you haven't used LaTeX before. For questions that require math derivations, you will need to TeX up your answers. Scanned versions of hand-written equations ARE NOT acceptable.

Note that pset0 is a simple un-graded homework that will help test your git, anaconda, and LaTeX setup. Please complete it as soon as possible, and contact course staff if you have any issues.

Staff / Office Hours

Staff

- **Instructor**: [Ayan Chakrabarti](#)
- **Course Staff**: Adith Boloor, Patrick Williams.

Office Hours: TBA. (will be through Zoom)