COMP 322 - Computer Vision and Image Processing

Fall 2026

Instructor	Hong Xu
Meetings	TR 9:30-10:45 am, Briggs 001
Course website	See Canvas
Email	xuh@rhodes.edu
Office	Briggs 210
Office hours	See Canvas for scheduled office hours; I am also available by appointment.
Textbook	Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. (free online on the author's website)
Prerequisite	COMP 241
CS Department Slack	See Canvas

Course Description

An introduction to computer vision, a discipline that develops methods to interpret and analyze images and videos. This course will focus heavily on computer vision as it relates to image processing, covering topics such as the characterization and representation of digital images, image enhancement, image restoration, image analysis, and image segmentation. We will also highlight classic computer vision topics such as object detection, three-dimensional reconstruction, tracking, and motion estimation. We will look at real-world applications of these techniques, including facial recognition, medical image analysis, robot navigation, and computational photography. The topics will be adjusted based on students' interests, so be prepared to share them.

Students should expect several substantial programming assignments requiring them to design, develop, test, and analyze computer vision and image processing programs. The course will also include labs, two midterm examinations, and a comprehensive final exam.

Course Objectives

By the end of this course, students should be able to:

- represent, manipulate, and analyze digital images using Python;
- implement common image processing operations, including filtering, enhancement, restoration, segmentation, and feature extraction;
- understand core computer vision problems such as detection, tracking, motion estimation, and 3D reconstruction at an introductory level;
- design, test, debug, and evaluate programs that operate on images and image-derived data;
- interpret the strengths and limitations of classical computer vision and image processing algorithms;
- communicate algorithmic choices, experimental results, and limitations clearly in code and written analysis.

Materials

The primary text is Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. The book is available free online. Additional readings, lecture notes, starter code, and assignment materials will be distributed through Canvas.

We will primarily use Python in this course. Students will use GitHub for source-code submission unless otherwise specified. Additional setup instructions will be provided through Canvas.

Coursework and Grading

Coursework	Weight
Labs	20%
Programming assignments	25%
Midterm exams	30%
Comprehensive final exam	25%

For borderline cases, I may take into account participation, attendance, improvement during the semester, and demonstrated engagement with the course.

Exam dates and the final exam time will follow the official Rhodes College schedule and will be announced on Canvas.

Grade Scale

Letter grade	Percentage range
A	93%-100%
A-	90%-92.99%
B+	87%-89.99%
B	83%-86.99%
B-	80%-82.99%
C+	77%-79.99%
C	73%-76.99%
C-	70%-72.99%
D	60%-69.99%
F	Below 60%

Workload

- It is important to stay current with the material. You should be prepared to devote regular weekly hours to this course, including substantial time outside of class.
- You should expect to spend significant preparation time working on a computer, trying example programs, developing assignments, debugging code, and analyzing results.
- Programming assignments are designed to be challenging and will often take multiple days to complete. Do not wait until the last minute to begin them.
- You are encouraged to form study groups to clarify and solidify your understanding of course concepts. However, unless an assignment is explicitly collaborative, submitted code and written solutions must represent your own effort.

Programming Assignments

- All programs assigned in this course must be written in Python unless otherwise specified.
- Programming assignments in this course are substantial. They will require careful design, implementation, testing, debugging, and analysis.
- The first line of each submitted source-code file should be a comment that states your name and the assignment number.
- Each student is responsible for keeping a backup copy of all source code submitted for an assignment.

- Source code must be uploaded to GitHub unless the assignment gives different submission instructions.
- Programs that do not run may receive a failing grade, so be sure to submit all required files and test your submission carefully.
- Late programs may be accepted with a penalty of one letter grade per day unless a genuine emergency arises. If you submit late work, email me so that I know to check the submission.
- Unless otherwise specified, programming assignments are to be completed individually.

Programming assignments will be graded using the following general guidelines:

- A-level work: The program is carefully designed, efficiently implemented, well documented, and produces clearly formatted, correct output.
- B-level work: The program works but may have minor or careless problems such as incomplete documentation, sloppy formatting, minor efficiency issues, weak testing, or small deviations from the assignment specification.
- C-level work: The program has more serious problems, including incorrect output for important cases, poor organization, inefficient design, incomplete documentation, or failure to follow important assignment requirements.
- D-level work: The program runs but produces clearly incorrect output for typical cases or deviates substantially from the assignment requirements.
- F-level work: The program does not run, produces little or no correct output, or does not meaningfully attempt the assignment.
- A grade of 0 may be assigned when no meaningful submission is made.

Rules for Completing Programming Assignments Independently

Unless otherwise specified, programming assignments handed in for this course are to be done independently. Talking to people is one of the best ways to learn, and I am willing to answer questions or provide hints if you are stuck. However, assistance crosses the line when it stops helping you learn and instead gives you the assignment solution.

Practical rule: In working on an assignment, you may not look at any correct program or correct piece of code for the same assignment that someone else has written.

- You may ask another student to help identify what is wrong with your incorrect code.
- Another student may suggest changes or explain a concept, but they may not write the needed code for you.
- You may look at examples from earlier assignments that use related techniques, provided they are not solutions to the current assignment.
- You may compare program outputs with classmates. Output is not program code.
- If outputs differ, you may not compare full programs to find the difference, since one of the programs may be correct.
- Sitting beside someone and typing what they have written is unacceptable.

AI-Assisted Tools

The computer science department defines the following five classes of AI usage:

1. **No AI Use Permitted**: Students are strictly prohibited from using any AI tools or resources for any aspect of assignments, projects, exams, or other course-related work. This includes, but is not limited to, content creation,idea generation, analysis, problem solving, or information retrieval.
2. **AI Use for Learning and Exploration Only (No Submission)**: Students may only use AI tools as a tutor to ask for explanations of concepts, methods and terminology. AI may not be used to generate any content used for submission. (This prohibition also applies to the submission of paraphrased or altered AI generated content.) All submitted work must be produced entirely by the student without copying from AI.

3. **AI Use for Specific Approved Tasks (Limited Use)**: AI tools are permitted for specific, clearly defined tasks within assignments, provided that the AI's contribution is explicitly cited and the student demonstrates a clear understanding of the AI-generated content.
4. **AI Use with Citation (Comprehensive Use)**: Students may use AI tools as a substantial aid in completing assignments. All AI assistance must be clearly cited, specific details documented, and the student must demonstrate a clear understanding of the AI-generated content.
5. **Unrestricted AI Use (Exploratory/Research Contexts)**: Students are free to use AI tools as they see fit to complete assignments and projects, with minimal or no restrictions on their application. The focus is on the outcome and the student's ability to achieve objectives. However, citation is encouraged.

For this class, different policies apply to different assessements:

Midterms/Final Exam: 1. No AI

Programming assignments and labs: 2. No Submission

Project: 4. Comprehensive use

Unless an assignment explicitly permits AI assistance, submitted code, analyses, and written answers must represent your own understanding and effort. You may not use AI tools to generate complete solutions, produce assignment code, or substitute for your own reasoning. If an assignment permits limited AI use, the permitted uses and required disclosure will be stated in that assignment. When in doubt, ask before using the tool.

Any submitted solutions to programming exercises, homeworks, labs, exams, and other graded items not wholly written by yourself or your group are considered plagiarism. All plagiarism will be reported to the Honor Council and will result in the grade of F for the course. This applies to AI use, copying, and any other form of plagiarism.

Exams

- There will be two midterm exams and a comprehensive final exam.
- Exams represent your own individual effort.
- Make-up exams will only be given in extreme circumstances.
- Exam dates, locations, and policies will be announced in class and on Canvas.

Class Conduct

- I encourage everyone to participate in class. Please do not be shy about asking questions; if you are confused about something, it is likely that someone else is confused as well.
- Teaching and learning are a partnership between the instructor and students. Asking questions helps you understand the material and helps me know what needs more explanation.
- Do not use your cell phone for calls or texting while in class, and keep ringers silenced.
- Use laptops and tablets for course-related work only.
- If you cannot make it to class, make sure you know what happened during the lecture you missed. It is your responsibility to catch up.

Attendance

- Attendance is expected for each class.
- Attendance will be checked each class period.
- Attendance, participation, and apparent overall improvement may be considered when assigning a final grade.
- After three unexcused absences, I reserve the right to reduce the final letter grade for the course by one letter grade for each additional absence.

Access and Accommodation

Your experience in this class is important to me. If you anticipate or experience physical or academic barriers based on disability, please let me know immediately so we can discuss options. If you have already established accommodations with Student Accessibility Services, please communicate your approved accommodations to me at your earliest convenience so we can discuss your needs in this course.

If you have not yet established services through Student Accessibility Services but have a condition that requires accommodations, please contact Student Accessibility Services. Rhodes College is committed to creating inclusive, equitable, and accessible learning environments for all students.

Academic Integrity

Plagiarism, cheating, and similar anti-intellectual behavior are serious violations of academic ethics and will be correspondingly penalized. If you are concerned about a possible violation of this kind, please talk with me. I would much rather have you turn in a partially completed assignment or do poorly on a test than violate the Rhodes Honor Code. I can, and very much want to, help you if you do not understand the material, but violations of academic integrity will be dealt with seriously.

Unless otherwise specified, everything you submit in this course must be your own work and represent your individual effort. Copying all or part of a solution, downloading a solution from the internet and submitting it as your own, having someone else provide the solution for you, or allowing someone else to copy from you are all reportable Honor Code violations. If you have any doubt about what behavior is acceptable, please ask me.

Diversity and Inclusion

A diverse learning community is a necessary element of a liberal arts education. Rhodes College is committed to fostering a community in which diversity is valued and welcomed. Discrimination or harassment on the basis of race, gender, color, age, religion, disability, sexual orientation, gender identity or expression, genetic information, national or ethnic origin, military status, or any other protected status will not be tolerated.

Freedom of thought, civil exchange of ideas, and appreciation of diverse perspectives are fundamental characteristics of a community committed to critical inquiry. Intellectual maturity requires individual struggle with unfamiliar ideas, and that challenge should take place in a climate of open-mindedness and mutual respect.

Sexual Misconduct Disclosure

I will do my best to help any student who comes to me with non-course-related concerns. Please keep in mind, however, that all faculty are required by policy to share knowledge of sexual assault, dating/domestic violence, sexual exploitation, stalking, sexual harassment, and sex/gender discrimination with the Title IX Coordinator. For more information about Rhodes' sexual misconduct policy or to make a report, please see www.rhodes.edu/titleix.

Tentative Course Topics

- Digital image representation and image formation
- Color spaces, sampling, quantization, and histograms
- Point operations and image enhancement
- Linear filtering, convolution, smoothing, and sharpening
- Edge detection and feature extraction
- Image restoration and noise models

- Segmentation and region-based methods
- Template matching and object detection
- Motion estimation, tracking, and optical flow
- Camera models and geometric vision
- Stereo and 3D reconstruction
- Applications in medical imaging, computational photography, robotics, and recognition

Syllabus Changes

I reserve the right to alter this syllabus as necessary. Any substantive changes will be announced in class and on Canvas. Any changes are intended to improve fairness, class quality, rigor, and learning to cater to student interest and progress as the semester goes on.