



(<http://toronto.edu/>)

CSC320: Introduction to Visual Computing

Fall 2025



photo credit (<https://vastphotos.com/>)

Course Description

This course is a first-principles introduction to the acquisition and computational processing of 2D images. It is aimed at undergraduates interested in learning about computer vision, digital photography and computer graphics.

The course serves as a stepping stone for tackling more advanced courses in those subjects and covers four broad themes:

- *Mathematical and engineering foundations*: introducing key concepts from geometry; multivariate calculus; linear algebra; image and signal processing; and human vision.

- *Algorithms for low-level computer vision*: image warping, morphing and stitching; image enhancement; image scissoring and inpainting; color image processing and display; face recognition; and 2D image matching.
- *Implementation*: implementing several such tools in Python and OpenCV.
- *A first taste of vision and graphics research*: understanding how to turn algorithmic descriptions in research papers into working computer vision code---and how to evaluate its performance---will be key skills acquired in the course.

Instructor



Aviad Levis

(<http://www.cs.toronto.edu/~alevis>)

Teaching Assistants



Esther Lin

(<https://estherlin.github.io/>)



Nhan (Len) Luong

(<https://lennemo09.github.io/>)



Ali Saraer Toosi

(<mailto:asaraert@dgp.toronto.edu>)



Andrew Xie

(<https://andrewxie.com/>)

Course Logistics

Lectures: Tuesdays 1:00pm-3:00pm in BA 1190.

Tutorials: Thursdays 1:00pm-2:00pm in FE 230.

Instructor office hour (week 2 onwards): Tuesday noon-1pm (BA7250) right before class.

TA office hour: TBD

Contact: Course announcements and general information will be posted Quercus. Q&A related to lectures, assignments and practice problems will take place on Piazza (<https://piazza.com/utoronto.ca/winter2025/csc320>).

Course dropbox: The one-stop shop for all course materials. See Quercus for the link.

Course Materials

Lecture Slides & Recordings

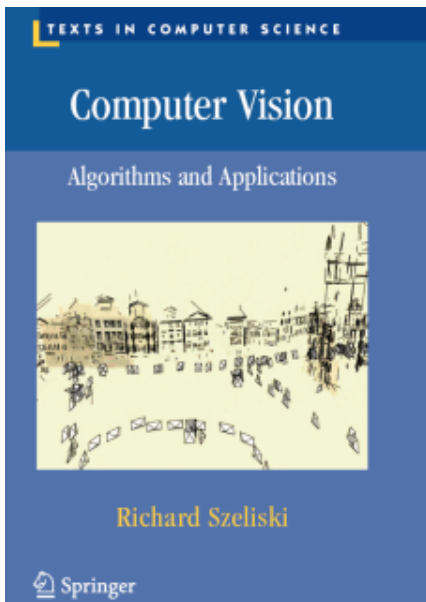
Links to slides for all lectures will be available on the course Dropbox approximately an hour before each lecture (see Quercus for Dropbox link). The full set of lecture slides from a recent offering of the course, taught by Kyros Kutulakos (<https://www.cs.toronto.edu/~kyros/courses/320/>), is already available on dropbox. This years course slides will be very similar to pervious term's offering so students who prefer reviewing the lecture content well in advance of a lecture can simply refer to those slides.

Recordings should be available on Quercus soon after each lecture. Since recording glitches are not uncommon due to issues with the system in specific classrooms (eg. missing sound, incomplete recordings, delays in posting) the completeness and timely availability of these recordings cannot be guaranteed, and they should not be treated as a substitute for in-person attendance.

Textbooks

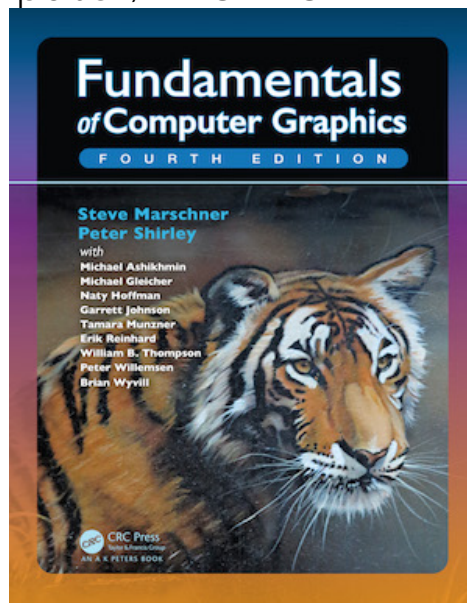
In addition to lecture slides, assigned readings consist of selected sections from the following textbooks. The vast majority of required readings come from the first two; the rest will be used very occasionally and/or only as optional readings for those interested in delving more deeply into the lecture's subjects. Several of these books are freely available online; the rest can be found in the library and/or are available for purchase. See the course schedule (<index.html#schedule>) for full details.

(<http://szeliski.org/Book/>)



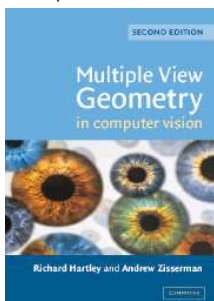
(<http://szeliski.org/Book/>)

(<https://www.oreilly.com/library/view/fundamentals-of-computer/9781482229417/>)



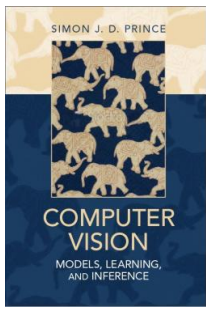
(<https://www.oreilly.com/library/view/fundamentals-of-computer/9781482229417/>)

(<https://www.robots.ox.ac.uk/~vgg/hzbook/>)



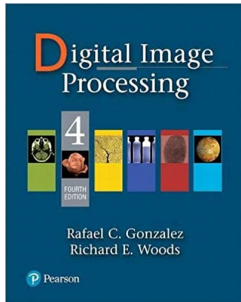
(<https://www.robots.ox.ac.uk/~vgg/hzbook/>)

(<http://www.computervisionmodels.com/>)



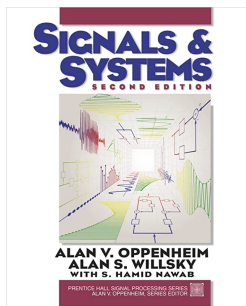
(<http://www.computervisionmodels.com/>)

(<https://www.pearson.com/en-us/subject-catalog/p/digital-image-processing/P200000003224/>)



(<https://www.pearson.com/en-us/subject-catalog/p/digital-image-processing/P200000003224/>)

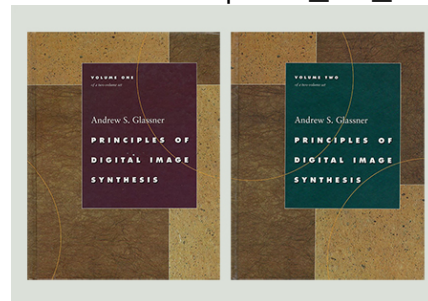
(<https://www.amazon.com/Signals-Systems-2nd-Alan-Oppenheim/dp/0138147574>)



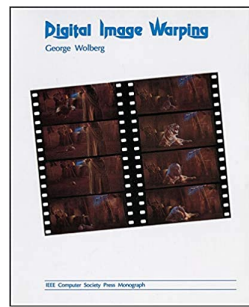
(<https://www.amazon.com/Signals-Systems-2nd-Alan-Oppenheim/dp/0138147574>)

(https://www.realtimerendering.com/Principles_of_Digital_Image_Synthesis)

(<https://www.wiley.com/en-us/Digital+Image+Warping-p-9780818689444>)



(https://www.realtimerendering.com/Principles_of_Digital_Image_Synthesis)



([https://www.wiley.com/en-](https://www.wiley.com/en-us/Digital+Image+Warping-p-9780818689444)

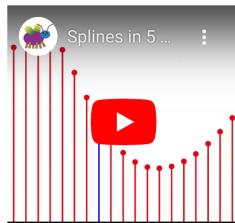
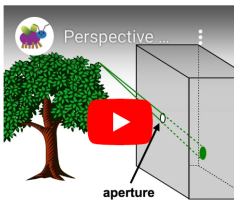
[us/Digital+Image+Warping-p-9780818689444](https://www.wiley.com/en-us/Digital+Image+Warping-p-9780818689444))

Videos

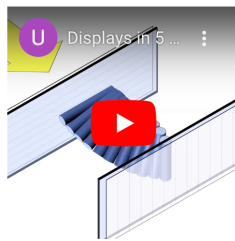
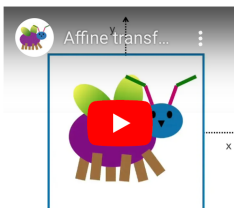
For the first half of the course, selected cartoon videos from Steve Seitz' "Graphics in 5 minutes (G5M)" (<https://g5m.cs.washington.edu/>) channel on youtube must be watched before each week's lecture. In addition to these videos, Shree Nayar's video lectures on the First Principles of Computer Vision (<https://fpcv.cs.columbia.edu/>) are an excellent source of lectures for optional viewing after class. See the course schedule (<index.html#schedule>) for full details.

(<https://g5m.cs.washington.edu/>)

Graphics in 5 Minutes



(<https://g5m.cs.washington.edu/>)



(<https://fpcv.cs.columbia.edu/>)

First Principles of Computer Vision

Lectures
Monographs
About
Research

Introduction

- Overview
- What is Computer Vision?
- What is Vision Used For?
- How Do Humans Do it?
- Topics Covered
- About the Lecture Series
- References and Credits

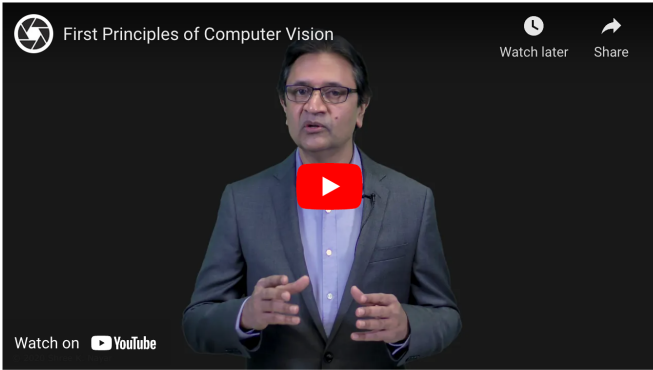
Imaging

Features

Reconstruction I

Reconstruction II

Perception



First Principles of Computer Vision

YouTube

(<https://fpcv.cs.columbia.edu/>)

Problem Sets & Past Exams

Solutions to all past term tests and final exams since 2016 can be found on the course Dropbox. The TAs have also compiled a selected set of practice problems from a variety of textbooks to test your knowledge of each lecture. See the course schedule (<index.html#schedule>) for full details.

Coursework

Assignments (60%)

There will be 4 programming assignments (see schedule). Each assignment comes with a partial implementation that you must complete, test, and write a report about. All except the first assignment will require reading an associated research paper and implementing (and testing) the technique it describes. Assignments will be exclusively in Python. All assignments will be posted on lecture day (Tuesdays) and will be due 3+ weeks later on the Tutorial day (Thursdays), with the exception of the last assignment, A4, that will be due on exactly 3 weeks later on Tuesday 02/12. Assignments are submitted via MarkUs (<https://markus.teach.cs.toronto.edu/markus/courses/45/>). The three best assignments will each count for 16% of the mark; the worst will count for 12%.

Late policy: All assignments are due 1 minute before midnight (11:59pm) on the due date. To account for unexpected technical last-minute issues with submissions, each student will be receive one-hour grace period for every assignment. For an extension of an assignment deadline (at most 7 days) please use the special considerations form (link available on Quercus). You must submit this form even if you have already made an absence declaration request on Acorn. Requests made over email to an instructor or TA will not be honored.

Academic honesty policy for written components: All reports submitted as part of your assignments are strictly individual work. No part of these reports should be shared with others, or taken from others. This includes verbatim text, paraphrased text, and/or images used. You are, however, allowed to discuss these components with others at the level of ideas, and indeed you are welcome and encouraged to brainstorm together.

Academic honesty policy for programming components: Collaboration on a programming component by individuals (whether or not they are taking the class) is encouraged at the level of ideas. Feel free to ask each other questions, brainstorm on algorithms, or work together on a (virtual or real) whiteboard. *Be careful, however, about copying actual code for programming assignments or merely adapting someone else's code.* This sort of collaboration at the level of artifacts is permitted if explicitly acknowledged, but will result in marks deduction. Full details about how to acknowledge such collaborations (and what deductions to expect) will be provided in the individual assignment handouts.

The use of generative artificial intelligence (AI) tools is strictly prohibited in all programming assignments. This includes, but is not limited to, ChatGPT, GitHub Copilot, and open-source models that you have trained and/or deployed yourself. You may not interact with, nor copy, paraphrase, or adapt any content from any generative AI for the purpose of completing programming assignments in this course. Use of generative AI will be considered use of an unauthorized aid, which is a form of academic misconduct under the Code of Behavior on Academic Matters. This course policy is designed to promote your learning and intellectual development and to ensure that our evaluations are a fair and accurate assessment of your learning. To that end, you will be required to include detailed comments along with the code you submit, and a random subset of students may be selected by a TA to explain their implementation in a live 1-1 meeting.

Exam (40%)

Final exam (no auto-fail): Date and time TBD.

Schedule and Syllabus

Lecture Week
Date
Description
G5M Videos to Watch Before Class
Readings & Exercises
Event
Deadline

Part I: Foundations

1
Tue 02/09
Course Intro Image Geometry 1 Homogeneous 2D coordinates, 2D transformations, homographies, image warping algorithms.
Affine transforms (https://www.youtube.com/watch?v=AheaTd_I5Is&feature=share&si=ELPmzJkDCLju2KnD5oyZMQ)
[materials01] (https://www.dropbox.com/scl/fi/ax99n14i4q76ump9audrr/materials_lec01.pdf?rlkey=u1m9848rsetqrxsfnfs3l3gyih&st=xl6qx094&dl=0)

Thu 04/09
Tutorial Image I/O in OpenCV and Python.

2
Tue 09/09
Image Geometry 2 Homogeneous 3D coordinates, pinhole imaging & perspective projection, imaging of 3D planes, panoramic image stitching, homography estimation.
Perspective projection (https://www.youtube.com/watch?v=F5WA26W4JaM&feature=share&si=ELPmzJkDCLju2KnD5oyZMQ), Perspective projection--the math (https://www.youtube.com/watch?v=g7Pb8mrwcJ0&feature=share&si=ELPmzJkDCLju2KnD5oyZMQ)
[materials02] (https://www.dropbox.com/scl/fi/mzenwssaxxfvnlhkce9yy/materials_lec02.pdf?rlkey=4tyurOa1bo827cg7wyk0tkcic&st=wii5ie9q&dl=0)
A1 out on Dropbox

Thu 11/09
Tutorial Linear algebra refresher: solving systems of linear equations. Lecture 1 practice problems.

3

Tue
16/09**Image Filtering 1**

Linear shift-invariant filters, convolution, basic filters.

Images (<https://www.youtube.com/watch?v=xUzhKqf22mY&feature=share&si=ELPmzJkDCLju2KnD5oyZMQ>),
 Image filtering (<https://www.youtube.com/watch?v=6v8dNtknOSM&feature=share&si=ELPmzJkDCLju2KnD5oyZMQ>)

[materials03] (https://www.dropbox.com/scl/fi/nj14q72kglw07yka1gezn/materials_lec03.pdf?rlkey=itimiguoh6uc1eq5lm0t9zur&st=ov0q5zgw&dl=0)

Thu
18/09**Tutorial**

Lecture 2 practice problems. A1 Q&A.

4

Tue
23/09**Image Filtering 2**

Gaussian filters, derivative filters, sharpening filters. The Bilateral filter. Discrete image formation, image interpolation.

Interpolation (<https://www.youtube.com/watch?v=Xj129kA3Ci0&feature=share&si=ELPmzJkDCLju2KnD5oyZMQ>)

[materials04] (https://www.dropbox.com/scl/fi/xuiv3jix4dcxshteybup5/materials_lec04.pdf?rlkey=injnd4ey0kobfnorqir3xy7c1&st=hb9hlmmm&dl=0)

Thu
25/09

Tutorial

Image filtering in action with OpenCV. Lecture 3 practice problems. Lecture 6 prep: Math refresher on complex numbers and their Euler representation

5

Tue
30/09

Image Filtering 3

Introduction to Fourier transforms, convolution theorem.

Fourier transforms (<https://www.youtube.com/watch?v=JciZYrh36LY&feature=share&si=ELPmzJkDCLju2KnD5oyZMQ>)

[materials05] (https://www.dropbox.com/scl/fi/ranqhc917q2rxbhnmkjjq/materials_lec06.pdf?rlkey=kqwa5dx8nf7nr5tmu49gimay5&st=ozri69sw&dl=0)

A2 out on Dropbox

Thu
02/10

Image Filtering 3 continued (no tutorial this week).

A1 due at 11:59pm

6
Tue 07/10
Image Filtering 4 Fourier-domain image filtering, hybrid images. Sampling theory basics, aliasing, anti-aliasing methods.
[materials06] (https://www.dropbox.com/scl/fi/epbdb70yz5f3yktbsz3mj/materials_lec07.pdf?rlkey=vfzpak4t3kppkpyem659zy5hc&st=c5jzyw95&dl=0)

Thu 09/10
Tutorial Lecture 3 & 4 practice problems. A2 overview and Q&A.

7
Tue 14/10
Color Vision & Camera ISPs Color perception in the human visual system, color imaging, color spaces, color displays. Smartphone Image Signal Processing pipelines.
Color (https://www.youtube.com/watch?v=6tTNgvAl1y4&feature=share&si=ELPmzJkDCLju2KnD5oyZMQ)
[materials07] (https://www.dropbox.com/scl/fi/4p4oy64ddfdtq1vbh9idp/materials_lec05.pdf?rlkey=0os2cexvqlxfbjlnrhvdvp26k&st=xk0me3w3&dl=0)

Thu
16/10

Tutorial

Fourier-domain filtering & anti-aliasing in action with OpenCV. A2 Q&A.

Part II: Image Representations

8

Tue
21/10

Differentiable 1D & 2D Representations

Representing smooth 2D curves, curve tangent, curve normal and the moving frame. Image gradient, image Laplacian, edge enhancement, Canny edge detection. Intelligent scissors, painterly rendering.

[no video]

[materials08] (https://www.dropbox.com/scl/fi/jhfvw3q4jf29ou8k20513/materials_lec08.pdf?rlkey=e77n5trm2d7jr9dnyq2s4aa9y&st=od0lknxa&dl=0)

A3 out on Dropbox

Thu
23/10

Tutorial

Lecture 7 practice problems. A3 Q&A.

A2 due at 11:59pm

28/10
30/10

Reading week break (No Lecture, Tutorial, Office hours)

9

Tue
04/11

Vector-Based Representations 1

Template matching, measuring image similarity, Harris/Foerstner corner detection. Introduction to PCA.

[no video]

[materials09] (https://www.dropbox.com/scl/fi/77tkmsn1eti0lb4qg05ve/materials_lec09.pdf?rlkey=b5cq2l5jfn52pq21e5ry9zytu&st=1sp4julg&dl=0)

Thu
06/11

Tutorial

Lecture 8 practice problems. Lecture 10 prep: Linear algebra refresher on matrix eigenvectors and eigenvalues.

10

Tue
11/11

Vector-Based Representations 2

Principal component analysis, face recognition using eigenfaces.

[no video]

[materials10] (https://www.dropbox.com/scl/fi/bqe588c0auyhzzam643y/materials_lec10.pdf?rlkey=ggx9petm2y4o7tig5fw0w5ysv&st=hw59buqy&dl=0)

A4 out on Dropbox

Thu
13/11

Tutorial

Lecture 9 practice problems. Overview of A4.

A3 due at 11:59pm

11

Tue

18/11

Multi-Scale Representations 1

The Discrete Haar Wavelet Transform, wavelet-based image compression, wavelet-based image processing.

[no video]

[materials11] (https://www.dropbox.com/scl/fi/cz5frOeoyg1gddiy8r1g6/materials_lec11.pdf?rlkey=ia7errr0b0apj9ubyo69wx2yv&st=1ck31jlv&dl=0)

Thu

20/11

Tutorial

Lecture 10 practice problems. A4 Q&A.

12

Tue

25/11

Multi-Scale Representations 2

Panoramic image stitching revisited. SIFT keypoints and SIFT-based correspondence-finding. Robust model fitting using RANSAC. Gaussian and Laplacian pyramids. Laplacian image blending.

[no video]

[materials12] (https://www.dropbox.com/scl/fi/tl5cg2t7ir361maynig6o/materials_lec12.pdf?rlkey=ptib6qd4jkm540n0ecct9ghfa&st=i4npqqtj&dl=0)

Thu 27/11
Tutorial Preparing for the final.

Tue 02/12
End of term
A4 due at 11:59pm

Additional Information

Prerequisites

- **Formal prerequisites:** Please consult the Arts and Science timetable (<https://artsci.calendar.utoronto.ca/course/csc320h1>) for full details on prerequisites and exclusions. Requests for prerequisite waivers should be directed to the Computer Science Undergraduate office (<mailto:cs.undergrad@utoronto.ca>)
- **Coding Experience:** All assignments will be in Python and will rely heavily on packages such as NumPy (<https://numpy.org/>) and OpenCV (<https://opencv.org/>). Advance knowledge of these packages is not necessary; they will be covered in tutorials as needed.
- **Course sequencing:** Starting in Fall 2024, CSC320 will be a prerequisite for CSC420 (Introduction to Image Understanding). Foundational material is being moved into CSC320 to make room for more advanced topics in CSC420. Students planning to take CSC317 (Computer Graphics) would also benefit from taking CSC320 first, or concurrently.

Hardware & Software Compatibility

- You are welcome to use your personal computer (OSX, Windows or Linux) for programming assignments. Be aware, however, that TAs will test your code on Teaching Labs computers and thus it is essential that you verify your code there before submission.
- We will support only one of the latest versions of python and recommend that you install it using Anaconda or Miniconda (see installation instructions here (<https://docs.anaconda.com/anaconda/install/>)). Further information about setting up your computing environment for the course's programming assignments will be given at the first tutorial and will also be available in the Tutorial 1 slides.

Related courses at UofT

- CSC317: Computer Graphics (<https://github.com/alecjacobson/computer-graphics-csc317>)
- CSC420: Introduction to Image Understanding (StG) (<https://www.cs.toronto.edu/~lindell/teaching/420/>)
- CSC2503: Foundations of Computer Vision (<https://www.cs.toronto.edu/~kyros/courses/2503/>)
- CSC2529: Computational Imaging (<https://www.cs.toronto.edu/~lindell/teaching/2529/>)
- CSC2530: Computational Imaging and 3D Sensing (<https://www.cs.toronto.edu/~kyros/courses/2530/>)
- CSC2539: Physics-Informed Neural Representations for Visual Computing (<https://www.cs.toronto.edu/~kyros/courses/2530/>)
- CSC2520: Geometry Processing (<https://github.com/alecjacobson/geometry-processing-csc2520>)

Acknowledgements

This version of the course was built (and taught in previous terms) by Kyros Kutulakos (<https://www.cs.toronto.edu/~kyros/>). The lecture slides have been revised significantly from offerings of the course prior to 2021 (<https://www.cs.toronto.edu/~kyros/courses/320/index.2018s.html>), and include slides from several other instructors. Feel free to use these slides for academic or research purposes, but please maintain all acknowledgments and source information. The format and style of this webpage are based on those of CS2529 (<https://www.cs.toronto.edu/~lindell/teaching/2529/>), taught by David Lindell. (<https://davidlindell.com/>)