UNIVERSITÉ Concordia UNIVERSITY

ACADEMIC YEAR 2018-2019
WINTER 2019

Course number	COMP 499G-691G	**Course Title**	Computer Vision
Department	Computer Science and Software Engineering	**Semester**	Winter 2019
Type	Elective	**Credits**	4.0
Level	Undergraduate	**Prerequisites**	very good programming knowledge/skills in C++ or Python is essential
Schedule	Tue @ 17:45 - 20:15 FG B070 SGW	**Lab/Tutorial**	Tu 8:30PM - 10:30PM, Tu 3:30PM - 5:30PM H967, H929
Instructor	Charalambos Poullis, CSSE	**Office hours**	Wed @ 14:00 - 15:00 and by appointment
Office	EV3.183	**Email**	charalambos@poullis.org
Teaching Assistants	Bodhiswatta Chatterjee Pijing [Alex] Xu	**Email**	computer.vision.ta@gmail.com

COURSE DESCRIPTION

This course introduces basic techniques and concepts in computer vision including image formation, grouping and fitting, geometric vision, recognition, perceptual organization, and the state-of-the art software tools. The student will learn fundamental algorithms and techniques, and gain experience in programming vision-based components; in particular, how to program in OpenCV, a powerful software interface used to process data captured from passive and active sensors.

COURSE OBJECTIVES

The primary objective of the course is to provide a comprehensive introduction to computer vision and the related programming principles required when designing and developing vision-based systems. More specifically the course aims to cover the following:

- Introducing the programming principles and algorithms used in computer vision
- Designing, developing, testing and debugging of vision-based components
- Learning modern GPU programming
- Gaining practical experience with the vision library, OpenCV

LEARNING OUTCOMES

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

- identify and explain the core concepts in computer vision
- employ programming principles, data-structures and algorithms of computer vision for processing
- compare, criticize and assess state-of-the-art techniques in computer vision
- develop OpenCV applications

COURSE CONTENTS

- Image formation: camera models, radiometry, linear filters and edge detection, interest point/feature detection
- Grouping and fitting: Hough transform, RANSAC, image alignment
- Geometric Vision: camera calibration, epipolar geometry, two-view and multi-view stereo, structure from motion
- Recognition: image classification, face detection and recognition, object detection, part-based models
- Perceptual organization: segmentation, optical flow
- A project
- Laboratory: two hours per week.

TEACHING METHOD

The course comprises of weekly lectures and practical training; both in the form of labs and individual assignments. **It is emphasized that attendance in lectures and labs is mandatory for learning and performing well in this course.**

ASSESSMENT

Assignments (x2)	2 × 10%
Quizzes (x2)	2 × 30%
Final project (x1)	1 × 20%

All assignments, quizzes and the final project must be completed in order to pass the class.

Quizzes: Quiz #1 will take place on **February 12th** and, Quiz #2 will take place on **April 02nd**.
Assignments/Final project: The goal of the programming assignments and the project is to gain practical experience in programming vision-based components with OpenCV. There are two programming assignments and one project. The assignments may have a small amount of extra credit. Please refer to the schedule for the assignment/project descriptions and due dates. All assignments must be done individually and ran during the lab session for evaluation. Lab instructors will test your knowledge of programming the assignment during this evaluation.
Submission: The assignments should be submitted by 17h00 on the day they are due. Late submissions will be accepted until 3 days following the initial deadline, however, there will be a penalty of 20% from the total assignment grade and [if applicable] no bonus. *The late submission policy does not apply to the project.*

ACADEMIC INTEGRITY POLICY

There is a plethora of online resources for OpenCV and Computer Vision in general. You are allowed to incorporate code or tips you find on the Web, provided this doesn't make the assignment/project trivial **and** you explicitly acknowledge your sources. You are allowed to discuss assignments with each other, but coding must be done individually.
Please make sure you familiarize yourself with Concordia's Academic Code of Conduct

SUGGESTED REFERENCE TEXTBOOKS

There is no prescribed textbook. Lecture slides for this course will be the primary pointers. There is a vast amount of learning content in the form of notes, programming tutorials, etc. available on the Internet. The following are suggested reference textbooks:

1. Computer Vision: A Modern Approach (2nd Edition) by David A. Forsyth, Jean Ponce. ISBN-13: 978-0136085928
2. Computer Vision: Algorithms and Applications by Richard Szeliski. ISBN-13: 978-1848829343. An electronic draft is available online: http://szeliski.org/Book/

Lecture notes will be provided for each lecture. These are a combination of multiple resources and materials generously made publicly available by S. Lazebnik, D. Forsyth, J. Ponce, J. Koenderink, S. Seitz, R. Szeliski, B. Freeman, M. Pollefeys, D. Lowe, K. Grauman, A. Efros, F. Durand, L. Fei-Fei, A. Torralba, R. Fergus, J. Hays.

COMMUNICATION

This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.
Find our class page here

COURSE SCHEDULE

The list below provides a summary of the material that will be covered during the course as well as a tentative schedule. Labs will support topics covered in the lectures and provide hands-on exercises.

Date	Lecture topic	Reading	Comments	Tutorial topic
01. Jan 8th	Course Overview Introduction to Computer Vision Images	Szeliski Ch. 1, 3.2 Forsyth/Ponce Ch. 4	Assignment 1 out	No tutorials
02. Jan 15th	Image sampling Edge detection	Szeliski Ch. 3.2, 3.4, 3.5, 4.2		setting up and filters (filter2D)
03. Jan 22nd	Geometric transformations Interest point operators	Szeliski Ch. 4.1.1	Assignment 1 due Assignment 2 out	tutorial_pyramids, edge detection
04. Jan 29th	Feature Descriptors Image Stitching I	Szeliski Ch. 4.1.2-4.1.3 Szeliski Ch. 6.1		Assignment 1 grading
05. Feb 05th	Image Stitching II Cameras	Szeliski Ch. 6 Szeliski Ch. 2	Assignment 2 due	Geometric transformations
06. Feb 12th	**QUIZ #1** **Assignment 1 Solution**			Q&A
07. Feb 19th	Multiple Views (stereo, epipolar geometry)	Szeliski Ch. 9	Project out	Assignment 2 grading
08. Mar 05th	Multiple views and motion (structure from motion, Motion and Optical Flow	Szeliski Ch. 7 Szeliski Ch. 8.4		Document scanner
09. Mar 12th	Sliding Window Face Detection with Viola-Jones Assignment 2 Solution	Szeliski Ch.14.1, 14.2		Generate depth image using stereo
10. Mar 19th	Image Classification Loss Function and Optimization			Face detection
11. Mar 26th	Back-propagation and Neural Networks Training Neural Networks		Project due	Q&A
12. Apr 02nd	**QUIZ #2**			
13. Apr 09th	Project solution			Project grading

GRADUATE ATTRIBUTES

As part of either the Computer Science or Software Engineering program curriculum, the content of this course includes material and exercises related to the teaching and evaluation of graduate attributes. Graduate attributes are skills that have been identified by the Canadian Engineering Accreditation Board (CEAB) and the Canadian Information Processing Society (CIPS) as being central to the formation of Engineers, computer scientists and information technology professionals. As such, the accreditation criteria for the Software Engineering and Computer Science programs dictate that graduate attributes are taught and evaluated as part of the courses. The following is the list of graduate attributes covered in this course, along with a description of how these attributes are incorporated in the course.
Knowledge base: Knowledge of computer vision. Introduction to vision API. Image formation: camera models, radiometry, linear filters and edge detection, interest point/feature detection. Grouping and fitting: Hough transform, RANSAC, image alignment. Geometric Vision: camera calibration, epipolar geometry, two-view and multi-view stereo, structure from motion. Recognition: image classification, face detection and recognition, object detection, part-based models. Perceptual organization: segmentation, optical flow.
Problem analysis: Use mathematical models as basis for the implementation of problems requiring computer vision. Analyze the requirements and constraints of the problem in order to determine what design and implementation solutions will be used.

Design: Design and compose computer vision components involving many aspects such as stated in the course description.
Use of tools: Use specific computer vision software development APIs to develop elaborated applications, make an educated decision on the tools and APIs to use based on the established requirements, constraints and design.