Department of Design & Computation Arts
CART 370: Real Time Video

Section AA

Thursdays   13:30-15:30   SGW EV-5.615   Class
Thursdays   15:30-17:30   SGW EV-5.815   Lab

Instructors: Sha Xin Wei, sha@alcor.concordia.ca
                     Freida Abtan, freida.abtan@gmail.com

Please make sure you indicate CART 370 in your email subject lines.

Office Hours: Wednesdays 15:30-17:00 SGW EV-6.769 or by appointment. Request for appointments should be made in person during class time or by email at least 48 hours in advance.

Course Website: http://hybrid.concordia.ca/~cart370_xinwei_11

Course Overview

Prerequisite: 24 credits in a Computation Arts, Electroacoustics, or Intermedia/Cyberarts program; or written permission of the Department.

CART 370 is a studio-based course in the creation and real-time processing of moving textures and video. This course surveys computer-based video art, particularly as applied to installation and performance arts practice. It provides an introduction to mathematical approaches for the real-time processing of 2D and higher-dimensional arrays, image and video filters, motion segmentation, and tracking blobs, optical flow, faces, and shapes.

NOTE: Students who have received credit for this topic under a CART 498 number may not take this course for credit.

NOTE: Students should have experience or knowledge in videography and video editing.

Class Policies

Attendance is required for this course. Unjustified absences will result in a loss of participation points. As per departmental policy, a maximum of 2 absences per term is tolerated, after which a medical certificate or documentation of another valid reason for the absences must be provided. Three unjustified absences per course will result in an automatic failure. Written notification will be sent to students after 2 missed classes.

Please arrive on time for class. Late students will loose participation points. Class arrivals that are late by more than 15 minutes will be considered as an absence for the offending student.

In all cases, students are responsible for any material they have missed through absence. Material from lectures and labs will not be repeated due to absence or tardiness. Exceptions to this policy might be made in the case of a legitimate medical (or other) emergency.

Students must turn off their cell phones during class time.

Assignments are due at the start of each class. Handing in assignments late will lower your grade by one full letter grade per day unless an extension is requested at least 24h in advance. Extensions will only be granted for reasonable excuses, such as an illness with a valid doctor's note.

The Department of Design and Computation Arts complies with the Code of Conduct of Concordia University. Students are expected to be familiar with it. Among other things, this code specifies that <u>plagiarism will not be tolerated in any form</u>. All patches (or algorithms) used by students in their assignments <u>must</u> be properly attributed by source, failure to do so counts as plagiarism as per university policy.

Deliverables

Assignments must be submitted to the CART370 class_shares folder on the fine arts hybrid server by the beginning of class time on the assignment due date. Each student should create a directory with their name, and submit all their assignment's patches and necessary included files within that directory. The patch to be executed to 'start' the assignment must be named start.maxpat. Specific and clear instructions to begin the patch must be included as documentation within the patches comments! A small rendered video of the program's output should also be included with each submitted video processing assignment, as should a text document that includes written documentation on your project's intention and implementation.

It is preferable that start.maxpat begins with a single 'bang' command, as my failure to operate your patch when marking will mean a reduced grade! Please remember that internal documentation is an important part of every patch. Not only is it specifically part of my marking scheme, it will it help you to implement your ideas clearly and will allow me to understand what you are trying to do when I examine your code afterward!

Resources

Cycling 74's Max, MSP, and Jitter Documentation includes fantastic tutorials on all aspects of the software. These are <u>highly</u> recommended, and many will be assigned as homework for the class.

All Patches demonstrated in class will be available from the course website. I will also maintain a useful list of externals and resources there.

Assignment 1: Movie Playback due week 4 - **September 29th**

Create a short video piece in Cycling74's Jitter, that makes use of at least 5 video files, or cues within video files. You can use any video manipulation objects that you would like, but must include at least 5 objects that begin with the jit prefix. Your work should be around 1 minute in length, and must demonstrate some kind of compositional or narrative development. Be prepared to discuss both your patch and your video piece in your assignment documentation. Your patch must execute with a single 'bang'.

Note: For source material, you may either shoot your own video, or use properly attributed video that has been published under the creative commons licensing agreement. You can also sample public works provided you follow the principles of fair use: if the work is unrecognizable, or if it samples small parts to make a completely different art object.

Assignment 2: Camera Input due week 6 - **October 13th**

Create a patch in Jitter, which receives live camera input and combines it with pre-rendered video to create something unique. You may take a narrative, visual, or conceptual approach to this assignment, but must be prepared to explain your artistic decisions, along with your programming, in your assignment documentation. Your patch should execute with a single 'bang' (after setting up the live camera feed) or contain explicit, simple instructions, on the steps necessary for the project to run, visible within the patch comments. You must also use at least 5 objects that begin with the jit. prefix that aren't from the quicktime library. Do not exceed use more than 90 seconds of pre-rendered video material in this project.

Note: For pre-rendered source material, you may either shoot your own video, or use properly attributed video that has been published under the creative commons licensing agreement. You can also sample public works provided you follow the principles of fair use: if the work is unrecognizable, or if it samples small parts to make a completely different art object.

Midterm Project due week 9 – **November 3rd**

For your midterm project, create a reactive video patch in Jitter. Your patch may respond to sound, light, color, or movement, some other definable attribute, or a combination of the above. It may incorporate pre-rendered video material that you have shot and edited yourself, live-camera input, and sound. You may take a narrative, visual, or conceptual approach to this assignment, however, you must be prepared to explain your artistic decisions as well as your reactive mapping strategies, along with the details of your programming implementation, in your assignment documentation. Be prepared to discuss all of these things when you present your midterm project to the class on the assignment due date. Your patch should execute with a single 'bang' (after setting up

any live camera feed) or contain explicit, simple instructions, on the steps necessary for the project to run, that are visible within the patch comments.

<u>Note</u>: If you choose to include sound material in this project, you may create your own, collaborate with someone else, or use properly attributed material that has been published under the creative commons licensing agreement.

Final Project due week 14 – **December 8th**

For your final project, you may work in groups of 2 and collaborate with outside artists, actors, dancers, musicians, models or performers to produce one of the following:

Option A: A site-specific installation
Option B: A multimedia production work (dance / theater / composition / other)
Option C: A significant video-processing tool.
Option D: A real-time audio-visual media performance.

Your project should include reactive elements and/or a substantial framework for media control and should reflect the material covered by this class. Your documentation must outline the significant details of your artistic intentions and implementation, as well as attribute responsibility within the group for the project's implementation. Video and written documentation are necessary for the project submission, as is a class presentation of your work. For projects that are location specific, or require performers who are not available to present within the context of the final class, your presentation may be composed of video documentation of an event at which the project was presented elsewhere, as well as an explanation of both the project and its programming implementation. As usual, your patch should execute with a single 'bang' (after setting up any live input devices) or contain explicit, simple instructions, on the steps necessary for the project to run, that are visible within the patch comments.

<u>Note</u>: If you choose to include sound material in this project, you may create your own, collaborate with someone else, or use properly attributed material that has been published under the creative commons licensing agreement.

Marking Scheme

Class Attendance and Participation	10
Jitter Assignment 1: Movie Playback	10
Jitter Assignment 2: Camera Input	15
Midterm Project	25
Final Project	40

Grading scale (for passing grades)

<u>Letter</u>	C+, C, C-	4.3, 4.0, 3.7
A+, A, A-	D+, D, D-	3.3, 3.0, 2.7
B+, B, B-	<u>G.P.A.</u>	2.3, 2.0, 1.7

| 1.3, 1.0, 0.7 | Outstanding | Satisfactory |
| Rating | Very Good | Marginal Pass |

Evaluation Criteria

Specific conceptual, technical and aesthetic objectives, associated with each individual assignment, will be clearly identified and evaluated by the following criteria:

- Theory (application, understanding)
- Programming (functionality, organization, design, documentation)
- Artistry (creativity, success, structure, palette)

Class Format

Lecture and Presentations	1:30 – 3:30
Break	3:30 – 3:45
In class work session	3:45 – 5:30

Class Schedule

Please note: This schedule is a departure point only, and is subject to change based on the background and aptitudes of the class.

Week 1: September 8th

Discussion of Syllabus
Introduction to Cycling74's Max
- How to Install Max and Documentation
- The Working Environment
- Max: Objects and Connections
- Inlets, Outlets, Flow of execution
- Mathematics Operators, Comparisons
- Class exercise: pseudo-code, incrementing counter
- How to download and install the trial version

Tutorial: Shooting video
Assignment: Max Tutorials 1 - 6

Week 2: September 15th

More max intro
- Sliders and other forms of input
- Routing, Gates and Switches
- Arguments
- Sub Patches

- Pack and unpack
- Class exercise: Visual Pattern from UI elements

Black Box programming
Digital video representation and storage
- Introduction to Video Matrixes
- jit.matrix, jit.unpack, jit.pack, jit.pwindow
- Playback/transport commands

Tutorial: capturing and editing video
Assignment: Max Tutorials 7 - 13, Jitter Tutorials 1 - 2

Week 3: September 22nd

Introduction to Jitter
- jit.window
- Exporting video files
- Cross-fading and editing

Programming in Max
- Storage and access
- Presets object
- Sequencing events

Class Exercise: Making a simple video sequencer

Lab: Re-sequencing a video
Assignment: Jitter Tutorials 3 - 5

Week 4: September 29th – Assignment 1 is due.

Sampling and Positioning
- jit.submatrix
- Scissors and Glue, multiplexing
- Downsampling
- Repositioning

Image Level Adjustment
- Brightness Contrast Saturation
- Luma and Hue
- Scale and Bias
- Introduction to Color space

Jitter Operators
Video Information and Math
Live camera feed

Lab: Creating a Video Mixer
Assignment: Jitter Tutorials 6 - 10

Week 5: October 6th

Spatial Transformations

Special Effects
Class Exercise: Revisiting the video sequencer
Mixing and Compositing
  • Advanced Cross Fading
  • Alpha blending
  • Luma and Chroma Keying
Video Generators
'Cleaning' video for processing

Lab: Faking an alpha channel

Week 6: October 13th – Assignment 2 is due.

Introduction to MSP
  • Sound file playback
  • DAC~ ADC~
  • Gain and Monitoring
  • Cycle~
  • Play~ Buffer~
  • Sfplay~
  • Groove~
  • Simple sound utilities
Class exercise: A sound file scrubber
Class exercise: Function based file playback
Class exercise: Feedback machine

Lab: Creating a sound 'instrument' (based on video drum machine example).

Week 7: October 20th

Extracting information
  • Statistics objects
  • Using color space information
Class exercise: creating a light Theremin
Value tracking
  • Jit.findbounds
  • Masking techniques
  • Pre-processing for tracking
Tracking change
  • Frame Differencing
  • Background subtraction

Lab: Mapping strategies for a sound instrument

Week 8: October 27th

Introduction to the Computer Vision Library
  • Installation

Special Effects
Class Exercise: Revisiting the video sequencer
Mixing and Compositing
  • Advanced Cross Fading
  • Alpha blending
  • Luma and Chroma Keying
Video Generators
'Cleaning' video for processing

Lab: Faking an alpha channel

Week 6: October 13th – Assignment 2 is due.

Introduction to MSP
  • Sound file playback
  • DAC~ ADC~
  • Gain and Monitoring
  • Cycle~
  • Play~ Buffer~
  • Sfplay~
  • Groove~
  • Simple sound utilities
Class exercise: A sound file scrubber
Class exercise: Function based file playback
Class exercise: Feedback machine

Lab: Creating a sound 'instrument' (based on video drum machine example).

Week 7: October 20th

Extracting information
  • Statistics objects
  • Using color space information
Class exercise: creating a light Theremin
Value tracking
  • Jit.findbounds
  • Masking techniques
  • Pre-processing for tracking
Tracking change
  • Frame Differencing
  • Background subtraction

Lab: Mapping strategies for a sound instrument

Week 8: October 27th

Introduction to the Computer Vision Library
  • Installation

- Motion-tracking objects
- Morphology objects

Temporal Processing
- Statistics objects
- jit.slide
- jit.matrixset

Lab: A visual response to motion

Week 9: November 3rd – Midterm Project is due.

Mid-term Presentations

Lab: Visit to the black box theatre space.
Demonstration of spatialization and compositing system
Projection considerations
- Surfaces
- Techniques

Week 10: November 10th

Introduction to the Computer Vision Library
- Blob-tracking
- Characteristic tracking
- Edge Detection
- Shape Information
- Pattern Recognition

Lab: Extracting Information from video

Week 11: November 17th

Math objects
- Linear algebra
- jit.expr
Revisiting spatial repositioning
- Seed video vs. Calculation
- Casting
Class Exercise: Calculating displacement
Lattice-based programming techniques

Lab: in-class work period. Each group must meet with the instructor to discuss their final project. Groups should be ready to pitch their project concept including their strategies for implementation.

Week 12: November 24th - Optional Course Topic

Introduction to using OpenGL in Jitter
- GPU vs. CPU
- Introduction to 3d objects
- Creating simple forms
- The render object
- Importing more complex objects
- Drawing Points and Lines
- Mapping video to a plane
- The slab object
- Compositing in OpenGL

Lab: In-class work period

Week 13: December 1st

In-class work period

Week 14: December 8th – Final Projects are due.

Final project presentations and critique