COMPSCI 373

- **Lecturers:**
  - A. Prof. Patrice Delmas (303.391) → Week 1-4
    - Contact details: p.delmas@auckland.ac.nz
      - Office: 303 - 391 (3rd level CompSci building)
    - Office hours: Tues-Thurs 3-4pm, particularly available right after each lecture
  - Dr Burkhard Wuensche (303.453) → Week 5-8
  - Prof. Georgy Gimel'farb (303.389) → Week 9-12
- **Tutors:**
  - Justin Nguyen, Trevor Gee
  - Check https://www.cs.auckland.ac.nz/courses/compsci373s1c/tutorials/
- **Lecture time:**
  - Tuesday 2-3pm (MLT 1)
  - Thursday 2-3pm (PLT 1)
  - Friday 2-3pm (Lib B15)
- **Marking:**
  - 25% assignments, 20% test, 55% examination
- **Assignments:**
  - 8.33 marks each (date out and due date online)
Information

- Cecil (https://www.cecil.auckland.ac.nz/) or LMS (https://lms.auckland.ac.nz/default.aspx)
  - To sit the Online multiple choice quizzes / programming quizzes
  - To check your marks
  - To download all lecture recordings
  - To download lecture notes, examples and exercises
- Forum (http://forums.cs.auckland.ac.nz) and Facebook page (semi-official)
  - For Announcement and FAQ
    □ Select COMPSCI 373 C S1 2014
- CS course page (http://www.cs.auckland.ac.nz/compsci373s1c)
  - To get information about our course

Assessments

- Note: Students **must** obtain a pass in both the practical (assignments) and non-practical work (test + exam) in order to pass as a whole
- Practical: (25%)
  - Assignments (8.33% each)
    □ A1 : due on Apr 9, 11:30pm
    □ A2 : due sometimes in May
    □ A3 : due on June 3, 11:30pm
- Theory: (75%)
  - Test: 20%,
    □ On Week 6 before Easter break
  - Exam: 55%
    □ TBA
Policy on Cheating and Plagiarism

- We use many ways to check that the work students submit for marking is their own and was not produced by, or copied from, someone else.
  - A comparison program to automatically compare all submissions from students.
- Note:
  - All assignments deemed to be too similar are automatically allocated a zero mark.
  - All students who submitted these assignments are entered in the duplicate assignment register (a list of students whose work is known to have been copied). This list is maintained over many years.
  - Students who have been caught cheating will be notified by email.
  - Offenders may be referred to the University Disciplinary Committee.
- Both the person who copied the work and the person whose work was copied are allocated a zero mark.
  - It is important that you do not lend your assignments to others. Never give anyone a copy of your assignment. It is the responsibility of each student to ensure that others do not copy their work.
  - Read: http://www.auckland.ac.nz/ua/home/about/teaching-learning/honesty

Assessments (con’t)

- Missed work
  - If you miss the deadline for an assignment and have a valid reason, you should see the course coordinator ASAP.
  - If you miss the test/exam for any valid reason, or you sit the test/exam but believe that your performance was impaired for some reason, then you may be able to apply for an aegrotat, compassionate or special pass consideration.
  - Refer to the online information on Missed Exams, Aegrotats and Compassionate Consideration: http://calendar.auckland.ac.nz/regulations/academic/examination.html
Part I Overview

Week 1: 2D and 3D Geometry

Week 2: Color Theory (Timo)

Week 3: Illumination and Shading

Week 4: Ray Tracing

References


C/C++:

- C Reference Manual
- Bruce Eckel - Thinking in C++
  http://www.ibiblio.org/pub/docs/books/eckel/
- C++ for Java Programmers
- See also CS373 resources page
A bit more about me

- 13 years at Uni (lecturing 105, 210, 373, 375, 716, 773, 775)
- What I dislike: noise during the class, cheating (test, exam, assignments)
- What I like: engaging students, questions from students, ask questions to students (mostly the ones sleeping, talking, texting or disrupting the class)
- What I do outside 105 this semester:
  - I teach 105 and 773
  - I do research in 3D computer vision:
    - [www.ivs.auckland.ac.nz/quick_stereo](http://www.ivs.auckland.ac.nz/quick_stereo)

Email me for
- Mistakes in the lecture notes
- Dead-links on the webpage
- Issues not related to courses (personal problems-valid request for extension)

Forum
- Check it as often as you want.
- I tend to read the forum (when close to assignment deadlines) and do answer but likely not as fast as the tutors.
- The tutors are taking care of the forum.

Email
- Check email the night before each lecture
  - There might be special tutorials
  - Other important information

Tutor
- Is here to help
- Has to help as many students as possible (be minded of others in need)
**Miscellaneous**

- We will have 3 lectures a week. You need to keep up with the pace:
  - Each new lecture will require you to know and understand the content of the previous lectures.
  - You must do the exercises provided with the lecture notes to make sure you really understand the course content.
  - Tutorials are a great way to supplement lectures.
  - Outside office hours and tutorials, tutors are not supposed to be at hand.

---

### How to progress while Learning

1. **Read the lecture notes after each lecture**
   - a. Make a summary of what has been seen on the lectures
   - b. Redo examples already treated (during the lectures) or/and do the untreated examples
   - c. Do examples without refereeing to the lectures

2. **Read the materials provided online**
   - a. To learn more
   - b. To complement lectures

3. **If you have questions or do not understand something**
   - a. Do 1
   - b. Do 2
   - c. Attend the tutorials
   - d. Check the forum
   - e. Ask other 373 students
   - f. Ask a tutor during office hours
   - g. Email me (only if point a to f are completed and did not bring relevant information)

4. **How to prepare exams**
   - a. Do previous years exams
   - b. Do exercises of the course/tutorials/exercise course book
   - ii. Do 1 and 2
A glimpse of the research at the Intelligent Vision Systems Lab, Department of computer Science, The University of Auckland

Low cost + advanced theory = improved performance

3 March 2014

Online Interactive Web-based Stereo Vision

- Build a portable/comprehensive website and mobile app:
  - Accept off-the-lab real-life photos of different types:
    - Left + right
    - Anaglyph
    - Auto-stereogram
  - Automatically align images to epipolar stereo form
  - Reconstruct 3D information and display online
  - Returns different 3D results
- URL: www.ivs.auckland.ac.nz/quick_stereo
- Thousands of visitors from 47 countries around the World
- First rank on Google Search on key words:
  - online stereo matching
  - online stereo vision
How to Get Digital Images?

1. 2D real $\rightarrow$ 2D digital
2. 3D real $\rightarrow$ 2D digital
3. 4D real $\rightarrow$ animated 2D digital
4. 3D real $\rightarrow$ 3D digital
5. 4D real $\rightarrow$ 4D digital
6. Synthetic or computer generated
What to Do with Digital Images?

- **Image Processing**: process them to get new images
- **Computer Vision**: analyze them to get information about what is in the image
- **Computer Graphics**: generate them

---

### OUTPUT

<table>
<thead>
<tr>
<th>INPUT</th>
<th>Descriptions</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptions</td>
<td></td>
<td>Computer Graphics</td>
</tr>
<tr>
<td>Images</td>
<td>Computer Vision</td>
<td>Image Processing</td>
</tr>
</tbody>
</table>

---

Pixels and Resolution

- **Pixel** or **pel** (picture element)
  - Position \((x,y)\) + signal value \(v\) (greyscale or colour)

  ![Position and Value](image)

  Position: (512, 456)
  Value: (10, 255, 8)

- **Resolution**: how many pixels? width \(\times\) height
- **Spatial resolution**: image pixels per cm or inch (in x and y)
  - Can be used to convert pixel coordinates into physical coordinates
Encoding of Colors

- **Bit-depth**: number of bits used to represent each pixel's value (typically 1, 8, 24 or 32)
- **Binary image**: bit depth is 1; only code values 0 (black) and 1 (white)
- **Scalar/monochrome/greyscale image**:
  - scalar code values (e.g. just a single number per color)
  - only grey values (from black to white) and no colour
- **Vector-valued image**:
  - vector code values (e.g. several numbers per color)
  - All the colors can be represented

Defining Images Mathematically

Images can be defined on an $M \times N$ arithmetic grid (or lattice)

$$\mathbb{R}_{M,N} = \{(x,y) : 1 \leq x \leq M \land 1 \leq y \leq N\}$$

- Pixel coordinates $x$ and $y$ with $x = 1, \ldots, M$; $y = 1, \ldots, N$
- Image as a function $f : \mathbb{R} \rightarrow \mathbb{V}$
- $\mathbb{V}$ is a set of signal values, e.g. grey levels or colors

**Example**:
- pixel at position (100, 100) has value 255, i.e. $f(100, 100) = 255$
Image Processing

- **Geometric transformations**: resizing, rotation, deformations, ...
- **Color transformations**: quantization, conversion, color adjustment, ...
- **Compositing**: combination of two or more images
- **Many other operations**

Computer Vision

- Use computer to do things similar to human vision, using image processing, artificial intelligence, biology & physics
- Usually dealing with 2D images of 3D scene
- Often real-time and part of a larger system (e.g. robot)
- E.g. scene reconstruction, image restoration, object recognition, tracking, motion estimation, event detection, …
Human vs. Computer Vision

**Human Vision**
- Subjective, unstable, inaccurate in measurements
- Involves active interaction with environment
- Exploits experience, knowledge, context
- But unique capabilities to describe and understand...
- Real-time

**Computer Vision**
- Objective accurate measurements
- But low capabilities to describe and understand...
- Rarely works effectively real-time
Generating Digital Images

Need a 3D scene consisting of:

- Objects (3D models)
- Light sources
- Camera (viewer, eye)
- Parameters specifying how light interacts with the scene
  - Light parameters
e.g. type of light (light bulb, spotlight), color, brightness
  - Material parameters
e.g. color, texture, transparency
  - Rendering parameters
e.g. algorithm, quality, resolution

Objects (3D Models)

Objects are made up of parts (often also of other objects):

- **Vertices**: 3D points used to define model
- **Edges**: lines between vertices
- **Faces**: polygons bounded by edges
- **Mesh**: surface made up of connected polygons
The Camera

- **View point**: where is the camera?
- **View direction**: which direction does it point?
- **View orientation**: where does the top of the camera point? E.g. is the photo taken upside down?
- **Projection**: what type of lens does the camera have?

**Rendering**:
- Project scene onto view plane in front of the camera
- Compute the color of each pixel on the view plane

---

**SUMMARY**
Summary

- Things we can do with images:
  - **Image Processing**: process them to get new images
  - **Computer Vision / Image Analysis**: analyze them to get higher-level information (context, emotional content, numbering/counting)
  - **Computer Graphics**: generate them

- Images:
  - pixels with encoded colors, (spatial) resolution, can be represented on arithmetic grid

- For generating images:
  - 3D scene with objects (vertices, edges, faces, meshes), light sources, camera

Quiz

1. What is the spatial resolution of an image?
2. What is Computer Vision?
3. What are the differences between Computer Vision and human vision?
4. What does a 3D scene consist of?
5. What are vertices, edges and faces?
Super Quiz

1. What is camera calibration?
2. How much time did each CG still shots of Gollum-LOTR take to be generated in 2001?
3. Same question for the Hobbit
4. What is the cheapest 3D camera commercially available if any?
5. How many 3D mobile phone do you know?
6. Cite 3 famous (most-used) software packages for image processing/computer graphics?
7. What was the price of a medium ranged 2Mpixels digital camera in 2002 compared to today’s 10 Mpixels camera?