COMPSCI 773

- Lecturers: Dr Patrice Delmas (731.312)
  A. Prof. Georgy Gimel'farb (731.320)
  A. Prof. John Morris (731.305)
- Lecture time: W 1.30pm-2.30pm F 12.30pm-2.30pm
- Marking:
  - 40% Final exam
  - 60% Lab work (30% group work, 20% individual assessment, 10% oral assessment → this might change depending on enrolment)
- 3 assignments

COMPSCI 773

1st assignment: image matching / stereo matching algorithms implementation, depth map construction, 3D visualisation (OpenGL, optional) – 03.04.09

2nd assignment: camera calibration, synchronisation of USB cameras for image acquisition, image pairs rectifications – 08.05.09

3rd assignment (06.06.09):
- 2+3D Face recognition (tentative)
- Whole system testing (live demo)

COMPSCI 773

- Individual assessment: (either or both)
  - Oral interview (with Patrice)
  - Research paper presentation during final 3 lectures (time permitting)
  - Overall involvement in group work
- Assignment reports (what's in):
  - A group report:
    - Who did what
    - What solution has been chosen and why
  - An individual report
    - Detailing parts each student did
    - Presenting OWN solution and results (if any)

COMPSCI 773

- The report should look like a research report with references
  - Justified explanations on the chosen solutions, graphs, results and
  - Critical assessment of the outcomes
- Programming
  - Windows C, C++ (Tentatively: Java for first assignment)
  - OpenGl, Gtk
  - You are allowed to use external libraries but you have to state it
  - You may be asked to pass your code to other groups for the next assignment
  - If you use another group's codes you MUST state it in your report

The project: Advanced biometrics: 2D/3D Face recognition

- Part 1:
  - Matching/stereo matching
  - Camera synchronisation
  - Database acquisition
  - Rectification
- Part 2: Face authentication
  - Goal: Identify faces from images using 2/3D data
  - 2D/3D Statistical analysis
  - Live face recognition
- Each group will have to do BOTH parts

What is available and what you will have to do

US: USB 2.0 web-cameras (2 per group)
- Calibration object
- 3D scanner
- PCs-Windows

YOU: A calibration object (*if you do not like ours*)
- Code in C, C++ (avoid Java) ➡ real-time
- Create a proper GUI integrating the different parts of the project
- Setup drivers if necessary
- A very strong team/personal effort

OUTCOME
- You will undertake work at the top-edge of today's research
- You will gain a unique experience of Applied Computer Vision

What is expected

Projects at the edge of research development: neither easy nor simple

- A great deal of work is required but: (1) you will learn a lot; (2) this could count as work experience in an expending IT area; (3) you can show what you are worth without having to fear too much about the final exam
  - Still 2 students failed because of exam in 2005, none since but some ended up with B-/C+ final grade
- The exam will encompass all that will be lectured **+** project-related questions
- It is not a good idea to concentrate on a very restricted part of the project as this will penalise you by the end
- If you like the projects you can continue towards the same directions for COMPSCI 780 / MSc studies
- If you think COMPSCI 773 is too hard for you we can offer a COMPSCI 780 project along the same directions

What you should know or learn very quickly

- C, C++ programming
- Confidence in mathematical skills (linear algebra)
- Basics of Image Processing
- To learn quickly (by yourself)
- GUI design (just the basics)
- A bit of OpenG for 3D display
- Camera control and synchronization
- The rest we will teach you…

Course contents

- Introductory lecture
- Image matching
- Stereo image matching
- 2D and 3D vision geometry
- Camera calibration
- Stereo calibration
- Segmenting binary images
- Feature extraction
- 3D scene description / understanding
- Real-time image processing
- Rectification of stereo pairs
- Colour discrimination
- Features classification: PCA-LDA
- Higher level statistical approaches
- Course overview and final demo

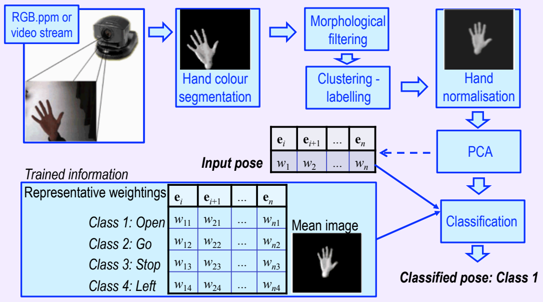
Project details: Part 1

- Matching / stereo matching
  - Implement a simple window-based correlation-based matching algorithm
- Acquire synchronised still streams from 2 webcams
  - Acquisition of face images
- Calibrate cameras
- Triangulate corresponding points to obtain their 3D positions
- Rectification
- Demo

Project details: Part 2

- Improve calibration accuracy
- Acquire face database with two web-cameras
- Extract faces from images
- Normalize database images
- Analyse 2D database images
- Compare new faces versus faces within database
- Repeat with 3D data
- Repeat with 2+3D data
- Create a GUI to interact with database
- Perform a real-time demo

A Possible Tentative Framework



Questions ?

- **First thing first**
  - Get access to the building 731 (one swipe card per group)
  - Get key of lab (one per group)
  - Create groups
- **Next**
  - I will do the first 4 weeks of the semester; Patrice - the 4 weeks following, John will do the last 4 weeks
  - **I am very keen to help and answer questions:**
    - Better ask from Wednesday after lectures to Friday
    - This is a research: I am keen to learn from you!
    - *My advice*: read research articles, respect assignments requirements, but feel very welcome to explore alternative solutions

Also, If Time Allows…

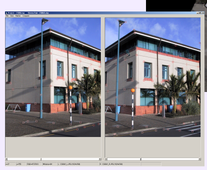
If not, then look through the following slides yourselves…

1. Introduction to some projects at Communication and Information Technology Research Group (CITR Tamaki)
   - Main research areas at CITR: see www.citr.auckland.ac.nz
2. Introduction to Image Processing
   - A few basic notions

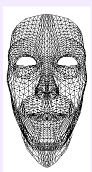
Main Research Areas at CITR

- Imaging and Image Technologies
  - 3D Shape Recovery and Computational Stereo Vision
  - Face and Gesture Recognition
  - 3D Face Analysis and Synthesis
  - Human Motion Estimation
  - Texture Analysis and Synthesis
  - Real-time Image Processing … *and many more*
- Data Communication
- Internet Programming

Binocular stereo

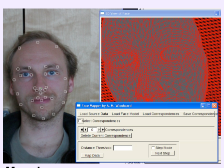


3D Stereo Reconstruction



3D model

2D image +



→ **3D face**

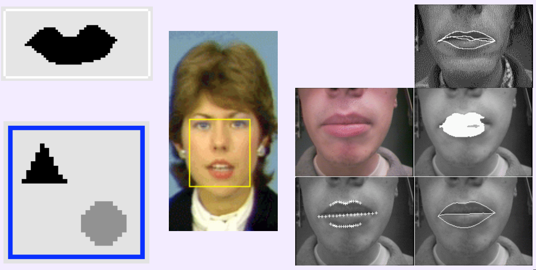
Mapping program

3D Face Rendering Using 2D Images

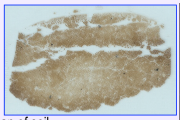
Face / Gesture Recognition

- Hand gesture recognition
  - Live demo for 7 different hand signs in our Active Vision Lab
- 3D face analysis and synthesis
  - Several systems setup (stereo, PSM, pattern projection, orthogonal views)
- Application of IP algorithms to face segmentation
  - Biometrics, Virtual Reality

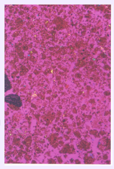
(Geodesic) Active Contours for Face Feature Extraction



Estimating Soil Properties Using Image Processing Techniques



Optical scan of soil core thin section



Electronic microscopy of soil core thin section

Dyed Soil Core Section

Results

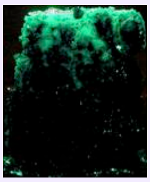
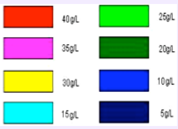
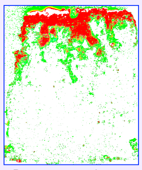


Image of a soil core cut vertically under black light illumination for 30 seconds

Segmented image:
White: area with dye
Black: area without dye

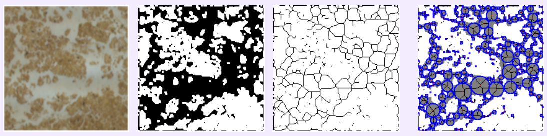




Dye concentration and flow pathways for a given soil core cut

Next steps:
- Reconstruct volume of flow pathways for a soil core
- Analyse variations over several soil cores
- Model the soil flow pathways

Using Mathematical Morphology



Original image
Blue: void
Brown: soil

Segmented image

Skeletons:
Line at maximal distance from void objects boundaries

Connected circles within the void network: this gives the connected maximal circular object within the void network

Goal: Characterise the void medium as it is related to soil properties with respect to fertilisers fate (it is related to soil pollution investigation)

Face Reconstruction Techniques

- **First project 2001**: Which technique to use for 3D face generation
  - Applications: VR, biometrics, etc
- Four different image processing techniques used in this project:
  - 3D Scanner
  - Shape from Shading
    - Photometric Stereo
  - Binocular Stereo
  - Structured Lighting

Lab Setup

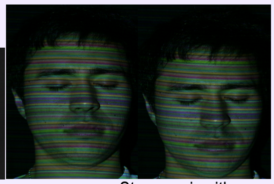
- 2 *Canon 10D EOS* reflex camera (8Mpixels)
- Optical bench with micrometric precision for epipolar alignment of cameras
- LCD projector (800×600 pixels)
- *Solutionix Rexcan 400 3D scanner* (sub-mm spacing, 300,000 points in less than a second)

How we proceed

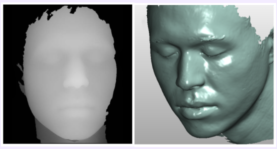
- Cameras are manually aligned (special procedure)
- Once cameras almost in epipolar position (almost aligned to a few pixel lines) we start the acquisition phase
- Acquisition:
  - First, a scan of the person: 2 *s*
  - LCD projectors project patterns (up to 12) and stereo images are acquired for each: 5 to 10 *s*
  - PSM procedure: 1 light - 1 image (L-R-M) then all lights on for texture: 2 *s*
  - Stereo pair of images: 1 *s*
- Processing: 3 PSM, 3 Structured Lighting, 9 stereo algorithms: Days…

Images

Stereo pair



Stereo pair with colour code projection



Ground truth

Results

Depth map without structured lighting

Depth map with structured lighting

Conclusion: SDPS + gray code is very close to ground truth

- 3D face database is currently built. Pb: Processing data
- Specific study on face feature accuracy

CITR

Towards a Low Cost Realistic Human Face Modelling and Animation Framework

Image and Vision Computing New Zealand ' 04

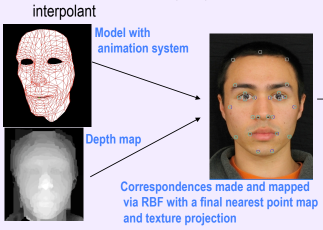
Presented by
Alexander Mark Woodward

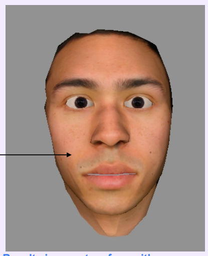
Supervisor: Dr. Patrice Delmas

• Feature extraction as a goal

Raw Data and Generic Model Interface

- User specifies a 'minimal' set of correspondences between raw and generic data
- Radial Basis Functions (RBF) are used as the interpolant

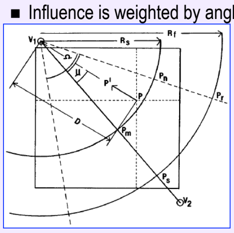




Results in a custom face with animation system in place

Linear Muscle

- Applies forces to nodes inside it's angular range
- Influence is weighted by angle and radius from muscle vector



Displacement formula:

$$\mathbf{p}' = \mathbf{p} + akr\frac{\mathbf{pv_1}}{\|\mathbf{pv_1}\|}$$

where $a = 1 - \frac{\cos(\mu)}{\cos(\Omega)}$;

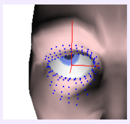
$$r = \begin{cases} \cos(1 - \frac{D}{R_s}\frac{\pi}{2}); & \text{for } \mathbf{p} \text{ inside sector } (\mathbf{v_1 p_n p_m p_1}) \\ \cos(\frac{D - R_s}{R_f - R_s}\frac{\pi}{2}); & \text{for } \mathbf{p} \text{ inside sector } (\mathbf{p_n p_r p_s p_m}) \end{cases}$$

and k - muscle contraction increment

Linear Muscle Contraction Video

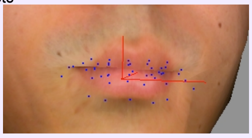
Ellipsoid Muscle

- Ellipsoid Muscle:
  - Acts like a string bag
  - Application of force weighted by radius only
  - Defined by major and 2 minor axes
  - Can generate puckering effects



Displacement formula:

$$\mathbf{p}' = \mathbf{p} + kr\frac{\mathbf{pv}_1}{\|\mathbf{pv}_1\|}$$



Ellipsoid Muscle Contraction Video

Expressions

- The animation system is now defined
- Contracting a muscle results in a reconfigured facial state
- An expression is thus a combination of muscle contractions
- Changing contraction coefficients over time achieves facial animation

Demo Video

Computer Vision: How to Mimic Human Visual Perception?

Dynamic and static 3D scenes, 2D images and video sequences, 2D visual patterns, …

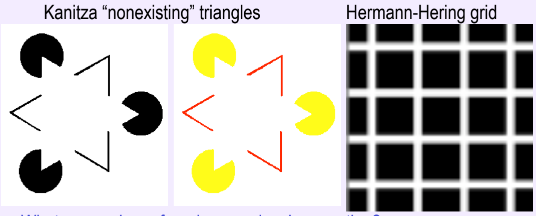
Human eye / brain:
Analysis and Understanding

Computer / camera(s):
Image acquisition / Active vision
Image processing
Image analysis / classification: object modelling, …
Image understanding

A Few Optical Illusions

- Examples from the MIT (MA, USA) website:

Kanitza "nonexisting" triangles — Hermann-Hering grid



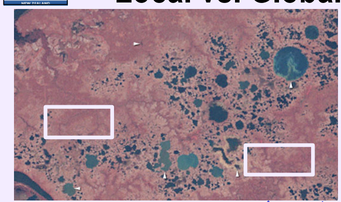
What can we learn from human visual perception?

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…
- **Computer vision:**
  - Objective accurate measurements
  - But low capabilities to describe and understand…

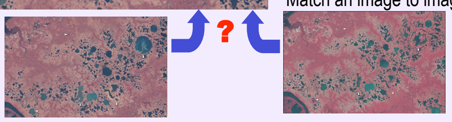
http://defendplanet.com/category/climate-change/

Local vs. Global Decisions



Match a part to the whole

Match an image to image



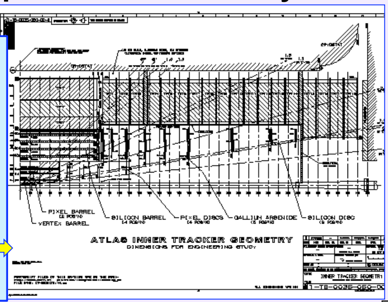
www.fys.uio.no/epf/atlas/poster.html

Example: Document Analysis

Signal / background

Noise elimination
Straight line detection
Circular line detection
(hard tasks if no prior knowledge / model !)

Basic elements
Syntax (relationships between elements)
Semantics (meaning of the document) ???

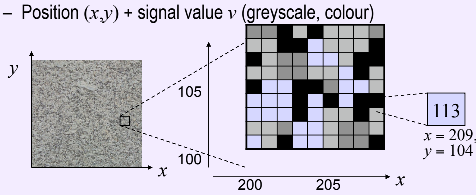


Digital Images

- Generally, 2D/3D/4D (time-varying 3D) images
  - Most of image processing courses consider 2D images
- A variety of ways to acquire a 2D image:
  - **2D digitisation** or **scanning**
    - Fax, paper / slide scanner, etc
  - **3D scene digitisation** or **direct acquisition**
    - Digital photo & video cameras, medical imaging: CT, MRI
  - **Synthetic** or **computer generated**
    - Computer graphics, virtual reality, animations, etc

TERMINOLOGY

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



  - Sometimes – only a position (x,y) within the image lattice
    - An image is supported by a finite lattrice (grid) with integer co-ordinates

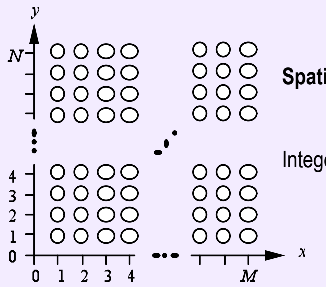
TERMINOLOGY

- **Spatial resolution**:
  - measure of the number of image pixels per unit length in horizontal or vertical direction, expressed in pixels / in, dots / in (dpi), samples / in, etc.
- **Monochrome** or **greyscale**
  - an image consisting of only grey values and no colour
- **Bit-depth**
  - the number of bits used to represent each pixel's value

TERMINOLOGY

- **Code value**
  - a pixel's value, e.g., in an 8-bit / pixel range of grey values from 0 (black) to 255 (white)
- **Binary image**
  - only code values 0 (black) and 1
- **Scalar image**
  - scalar code values (e.g. greyscale image)
- **Vector-valued image**
  - vector code values (e.g. 3-channel colour image)

Arithmetic Grid



Spatial pixel coordinates
$\mathbf{p} = (x,y)$

Integer values
$x = 1,\dots,M;$
$y = 1,\dots,N$

Arithmetic Grid

- $M \times N$ arithmetic grid (or lattice):
  $$\mathbf{R}_{M,N} = \{(x,y): 1 \le x \le M \wedge 1 \le y \le N\}$$
- Cartesian coordinates x_c, y_c relate to the integer pixel coordinates x,y as
  $$x_c = x_0 + x/s_x \,;\, y_c = y_0 + y/s_y$$
  where s_x, s_y denote spatial resolution measured in e.g. pixels per inch or pixels per mm

Digital Image

- **Image model**: an image as a function f: $\mathbf{R} \to \mathbf{V}$
  - $\mathbf{V}$ is a set of signal values (grey levels, colours, …)
  - **Intensity** (grey level): $f(\mathbf{p}) \equiv f(x,y) \in \mathbf{V} = \{0, 1, \dots, G_{\max}\}$;
    - Typically $G_{\max} = 255$, or 8 bit per pixel
  - **Colour:** three components, e.g., **R**ed, **G**reen, **B**lue levels $\mathbf{f}(\mathbf{p}) \equiv \mathbf{f}(x,y) \in \mathbf{V}^3$
    - $\mathbf{f}(x,y) = [255, 0, 0]$
    - $\mathbf{f}(x,y) = [0, 255, 0]$
    - $\mathbf{f}(x,y) = [0, 0, 255]$

Pixel: (x,y,**v**=**f**(x,y))

Image Pyramids

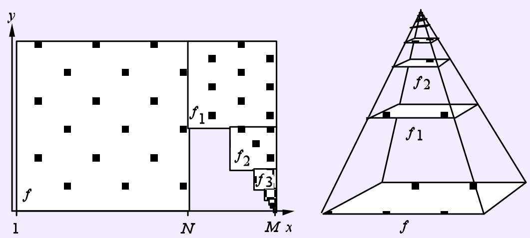


Image Pyramids