CreativeAI
Deep Learning for Graphics

Niloy Mitra
UCL

Iasonas Kokkinos
UCL

Paul Guerrero
UCL

Nils Thuerey
TUM

Tobias Ritschel
UCL

http://geometry.cs.ucl.ac.uk/creativeai/
People

Niloy Mitra
People

Niloy Mitra

Iasonas Kokkinos
People

Niloy Mitra  Iasonas Kokkinos  Paul Guerrero
People

Niloy Mitra  Iasonas Kokkinos  Paul Guerrero  Nils Thuerey
People

Niloy Mitra  Iasonas Kokkinos  Paul Guerrero  Nils Thuerey  Tobias Ritschel
People

Niloy Mitra
Iasonas Kokkinos
Paul Guerrero
Nils Thuerey
Tobias Ritschel
## Timetable

<table>
<thead>
<tr>
<th>Topic</th>
<th>Time</th>
<th>Niloy</th>
<th>Paul</th>
<th>Nils</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>2:15 pm</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Machine Learning Basics</td>
<td>~ 2:25 pm</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural Network Basics</td>
<td>~ 2:55 pm</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Feature Visualization</td>
<td>~ 3:25 pm</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Alternatives to Direct Supervision</td>
<td>~ 3:35 pm</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>15 min. break</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image Domains</td>
<td>4:15 pm</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3D Domains</td>
<td>~ 4:45 pm</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Motion and Physics</td>
<td>~ 5:15 pm</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Discussion</td>
<td>~ 5:45 pm</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Code Examples

PCA/SVD basis
Linear Regression
Polynomial Regression
Stochastic Gradient Descent vs. Gradient Descent
Multi-layer Perceptron
Edge Filter ‘Network’
Convolutional Network
Filter Visualization
Weight Initialization Strategies
Colorization Network
Autoencoder
Variational Autoencoder
Generative Adversarial Network

http://geometry.cs.ucl.ac.uk/creativeai/
Course Objectives
Course Objectives

• Provide an overview of the popular ML algorithms used in CG
Course Objectives

• Provide an overview of the popular ML algorithms used in CG

• Provide a quick overview of theory and CG applications
  • Many extra slides in the course notes + example code
Course Objectives

• Provide an overview of the popular **ML algorithms** used in CG

• Provide a quick overview of **theory** and **CG applications**
  • Many extra slides in the course notes + example code

• Progress in the last 3-5 years has been dramatic
  • We have organized them to help newcomers
  • Discuss the main **challenges and opportunities** specific to CG
Two-way Communication
Two-way Communication

• Our aim is to convey what we found to be relevant so far

• You are invited/encouraged to give feedback
Two-way Communication

• Our aim is to convey what we found to be relevant so far

• You are invited/encouraged to give feedback
  • Speakup. Please send us your criticism/comments/suggestions
Two-way Communication

• Our aim is to convey what we found to be relevant so far

• You are invited/encouraged to give feedback
  • Speakup. Please send us your criticism/comments/suggestions
  • Ask questions, please!

• Thanks to many people who helped so far with slides/comments
Representations in CG
Representations in CG

• Images (e.g., pixel grid)

• Volume (e.g., voxel grid)
Representations in CG

• Images (e.g., pixel grid)

• Volume (e.g., voxel grid)

• Meshes (e.g., vertices/edges/faces)
Representations in CG

• Images (e.g., pixel grid)

• Volume (e.g., voxel grid)

• Meshes (e.g., vertices/edges/faces)

• Animation (e.g., skeletal positions over time; cloth dynamics over time)
Representations in CG

• Images (e.g., pixel grid)

• Volume (e.g., voxel grid)

• Meshes (e.g., vertices/edges/faces)

• Animation (e.g., skeletal positions over time; cloth dynamics over time)

• Pointclouds (e.g., point arrays)
Representations in CG

• Images (e.g., pixel grid)

• Volume (e.g., voxel grid)

• Meshes (e.g., vertices/edges/faces)

• Animation (e.g., skeletal positions over time; cloth dynamics over time)

• Pointclouds (e.g., point arrays)

• Physics simulations (e.g., fluid flow over space/time, object-body interaction)
Problems in Computer Graphics

• Feature detection (image features, point features) \( \mathbb{R}^{m \times m} \rightarrow \mathbb{Z} \)

• Denoising, Smoothing, etc. \( \mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m} \)

• Embedding, Distance computation \( \mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^{d} \)

• Rendering \( \mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m} \)

• Animation \( \mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m} \)

• Physical simulation \( \mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m} \)

• Generative models \( \mathbb{R}^{d} \rightarrow \mathbb{R}^{m \times m} \)
## Problems in Computer Graphics

<table>
<thead>
<tr>
<th>Feature detection (image features, point features)</th>
<th>$\mathbb{R}^{m \times m} \rightarrow \mathbb{Z}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denoising, Smoothing, etc.</td>
<td>$\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$</td>
</tr>
<tr>
<td>Embedding, Distance computation</td>
<td>$\mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^{d}$</td>
</tr>
<tr>
<td>Rendering</td>
<td>$\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$</td>
</tr>
<tr>
<td>Animation</td>
<td>$\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$</td>
</tr>
<tr>
<td>Physical simulation</td>
<td>$\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$</td>
</tr>
<tr>
<td>Generative models</td>
<td>$\mathbb{R}^{d} \rightarrow \mathbb{R}^{m \times m}$</td>
</tr>
</tbody>
</table>
# Problems in Computer Graphics

- **Feature detection (image features, point features)**
\[ \mathbb{R}^{m \times m} \rightarrow \mathbb{Z} \]

- **Denoising, Smoothing, etc.**
\[ \mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m} \]

- **Embedding, Distance computation**
\[ \mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^{d} \]

- **Rendering**
\[ \mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m} \]

- **Animation**
\[ \mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m} \]

- **Physical simulation**
\[ \mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m} \]

- **Generative models**
\[ \mathbb{R}^{d} \rightarrow \mathbb{R}^{m \times m} \]
Goal: Learn a Parametric Function

\[ f_\theta : \mathbb{X} \rightarrow \mathbb{Y} \]

\(\theta\): function parameters, \(\mathbb{X}\): source domain, \(\mathbb{Y}\): target domain; these are learned
Goal: Learn a Parametric Function

\[ f_\theta : \mathbb{X} \longrightarrow \mathbb{Y} \]

\( \theta \): function parameters, \( \mathbb{X} \): source domain \( \mathbb{Y} \): target domain

these are learned

Examples:
Goal: Learn a Parametric Function

\[ f_\theta : \mathbb{X} \rightarrow \mathbb{Y} \]

\( \theta \): function parameters, \( \mathbb{X} \): source domain \( \mathbb{Y} \): target domain
these are learned

Examples:

Image Classification: \( f_\theta : \mathbb{R}^{w \times h \times c} \rightarrow \{0, 1, \ldots, k - 1\} \)
\( w \times h \times c \): image dimensions \( k \): class count
Goal: Learn a Parametric Function

\[ f_\theta : \mathbb{X} \rightarrow \mathbb{Y} \]

\( \theta \): function parameters, \( \mathbb{X} \): source domain \( \mathbb{Y} \): target domain

these are learned

Examples:

**Image Classification:**

\[ f_\theta : \mathbb{R}^{w \times h \times c} \rightarrow \{0, 1, \ldots, k - 1\} \]

\( w \times h \times c \): image dimensions \( k \): class count

**Image Synthesis:**

\[ f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^{w \times h \times c} \]

\( n \): latent variable count \( w \times h \times c \): image dimensions
Machine Learning 101: **Linear Classifier**

Each data point has a class label:

$$y^i = \begin{cases} 
1 & (\bullet) \\
0 & (\circ) 
\end{cases}$$

$$f_\theta : \mathbb{R}^n \rightarrow \{0, 1\}$$
Machine Learning 101: Linear Classifier

Each data point has a class label:

\[ y^i = \begin{cases} 
1 & (\bullet) \\
0 & (\circ) 
\end{cases} \]

\[ f_\theta : \mathbb{R}^n \rightarrow \{0, 1\} \]
Machine Learning 101: Linear Classifier

Each data point has a class label:

\[
y^i = \begin{cases} 
1 & \text{for red points} \\
0 & \text{for blue points}
\end{cases}
\]

\[
f_\theta : \mathbb{R}^n \rightarrow \{0, 1\}
\]

\[
f_\theta(x) = \begin{cases} 
1 & \text{if } wx + b \geq 0 \\
0 & \text{if } wx + b < 0
\end{cases}
\]
Machine Learning 101: **Linear Classifier**

Each data point has a class label:

$$f_\theta : \mathbb{R}^n \rightarrow \{0, 1\}$$

$$f_\theta(x) = \begin{cases} 
1 & \text{if } wx + b \geq 0 \\
0 & \text{if } wx + b < 0 
\end{cases}$$

$$\theta = \{w, b\}$$

$$y^i = \begin{cases} 
1 & (\text{•}) \\
0 & (\text{○}) 
\end{cases}$$
Data-driven Algorithms (Supervised)

Labelled data
(supervision data)
Data-driven Algorithms (Supervised)

Labelled data (supervision data) → ML algorithm → Trained model
Data-driven Algorithms (Supervised)

Labelled data (supervision data) → ML algorithm → Trained model

Test data (run-time data)
Data-driven Algorithms (Supervised)

Labelled data (supervision data) → ML algorithm → Trained model → Prediction

Test data (run-time data)
Data-driven Algorithms (Supervised)

Labelled data (supervision data) → ML algorithm

Trained model

Validation data (supervision data)

converged?

Test data (run-time data) → Trained model → Prediction
Data-driven Algorithms (Supervised)

Labelled data (supervision data) → ML algorithm → converged?

Validation data (supervision data)

Test data (run-time data) → Trained model → Prediction
Data-driven Algorithms (Supervised)

- Labelled data (supervision data) → ML algorithm → converged? → Validation data (supervision data)
- Test data (run-time data) → Trained model → Prediction

Implementation Practice: Training: 70%; Validation: 15%; Test 15%
Data-driven Algorithms (Unsupervised)

Training data → ML algorithm → converged? → Validation data

Test data (run-time data) → Trained model → Prediction

Implementation Practice: Training: 70%; Validation: 15%; Test 15%
Various ML Approaches (Supervised approaches)

Various ML Approaches (Supervised approaches)

Various ML Approaches (Supervised approaches)

Various ML Approaches (Supervised approaches)

<table>
<thead>
<tr>
<th>Input data</th>
<th>Nearest Neighbors</th>
<th>Linear SVM</th>
<th>RBF SVM</th>
<th>Gaussian Process</th>
<th>Decision Tree</th>
<th>Random Forest</th>
</tr>
</thead>
</table>

Various ML Approaches (Supervised approaches)

Rise of Learning

• 1958: Perceptron
• 1974: Backpropagation
• 1981: Hubel & Wiesel wins Nobel prize for ‘visual system’
• 1990s: SVM era
• 1998: CNN used for handwriting analysis
• 2012: AlexNet wins ImageNet
Rise of Machine Learning

- neural network
- artificial intelligence
- machine learning
Rise of Machine Learning

- neural network
- artificial intelligence
- machine learning

Note
Rise of Machine Learning

- neural network
- artificial intelligence
- machine learning
Rise of Machine Learning

- neural network
- artificial intelligence
- machine learning
Rise of Machine Learning (in Graphics)

- Machine learning
- Neural network

Graphs showing the increase in machine learning and neural network topics from 2013 to 2017 in SIG+SA+EG+SGP+EGSR and Eurographics.
What is Special about CG?
What is Special about CG?

1. **Image Processing** (image translation tasks)
What is Special about CG?

1. **Image Processing** (image translation tasks)

2. Many sources of input data — **model building** (e.g., images, scanners, motion capture)
What is Special about CG?

1. **Image Processing** (image translation tasks)

2. Many sources of input data — **model building**
   (e.g., images, scanners, motion capture)

3. Many sources of **synthetic data** — can serve as supervision data
   (e.g., rendering, animation)
What is Special about CG?

1. **Image Processing** (image translation tasks)

2. Many sources of input data — **model building**
   (e.g., images, scanners, motion capture)

3. Many sources of **synthetic data** — can serve as supervision data
   (e.g., rendering, animation)

4. Many problems in **generative models**
Main Challenges and Scope for Innovation
Main Challenges and Scope for Innovation

1. **Representation**: How is the data organised and structured?
Main Challenges and Scope for Innovation

1. **Representation**: How is the data organised and structured?

2. **Training data**: Is it synthetic or real, or mixed?
Main Challenges and Scope for Innovation

1. **Representation**: How is the data organised and structured?

2. **Training data**: Is it synthetic or real, or mixed?

3. **User control**: End-to-end or in small steps?
Main Challenges and Scope for Innovation

1. **Representation**: How is the data organised and structured?

2. **Training data**: Is it synthetic or real, or mixed?

3. **User control**: End-to-end or in small steps?

4. **Loss functions**: Hand-crafted or learned from data?
End-to-end: Learned **Features**
End-to-end: Learned **Features**

- **Old days**
  - Handcrafted feature extraction, e.g., edges or corners (hand-crafted)
  - Mostly with linear models (PCA, etc.)
End-to-end: Learned Features

• **Old days**
  • Handcrafted feature extraction, e.g., edges or corners (hand-crafted)
  • Mostly with linear models (PCA, etc.)

• **Now**
  • End-to-end
  • Move away from hand-crafted representations
End-to-end: Learned Loss
End-to-end: Learned Loss

• *Old days*
  • Evaluation came after
  • It was a bit optional
    • You might still have a good algorithm without a good way of quantifying it
    • Evaluation helped publishing
End-to-end: Learned **Loss**

- **Old days**
  - Evaluation came after
  - It was a bit optional
    - You might still have a good algorithm without a good way of quantifying it
    - Evaluation helped publishing

- **Now**
  - It is essential and build-in
  - If the loss is not good, the result is not good
  - (Extensive) Evaluation happens automatically
End-to-end: Learned **Loss**

- **Old days**
  - Evaluation came after
  - It was a bit optional
    - You might still have a good algorithm without a good way of quantifying it
    - Evaluation helped publishing

- **Now**
  - It is essential and build-in
  - If the loss is not good, the result is not good
  - (Extensive) Evaluation happens automatically

- While still much is left to do, this makes graphics much more reproducible
End-to-end: Real/Generated Data
End-to-end: Real/Generated Data

- **Old days**
  - Test with some toy examples
  - Deploy on real stuff
  - Maybe collect some performance data later
End-to-end: Real/Generated Data

- **Old days**
  - Test with some toy examples
  - Deploy on real stuff
  - Maybe collect some performance data later

- **Now**
  - Test and deploy need to be as identical *(in distribution)*
  - Need to collect data first
  - No two steps
Examples in Graphics

- Geometry
- Image manipulation
- Rendering
- Animation
Examples in Graphics

Geometry
- Mesh segmentation
- Procedural modelling
- Learning deformations

Animation
- Animation
- Boxification
- Facial animation
- PCD processing

Rendering
- Image manipulation
- Sketch simplification
- Colorization
- BRDF estimation
- Real-time rendering
- Denoising

Geometry
Examples in Graphics

- Sketch simplification
- Colorization
- Procedural modelling
- Mesh segmentation
- Learning deformations
- Real-time rendering
- BRDF estimation
- Animation
- Boxification
- Denoising
- Fluid
- Facial animation
- PCD processing
Course Information (slides/code/comments)

http://geometry.cs.ucl.ac.uk/creativeai/