CreativeAI: Deep Learning for Computer Graphics

Supervised Learning in CG

Niloy Mitra  Iasonas Kokkinos  Paul Guerrero  Vladimir Kim  Nils Thuerey  Leonidas Guibas

UCL/Adobe  UCL/Ariel AI  UCL/Adobe  Adobe  TU Munich  Stanford University/FAIR
# Timetable

<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
<th>Niloy</th>
<th>Iasonas</th>
<th>Paul</th>
<th>Nils</th>
<th>Leonidas</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00</td>
<td>Introduction</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>~9:15</td>
<td>Neural Network Basics</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>~9:50</td>
<td>Supervised Learning in CG</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>~10:20</td>
<td>Unsupervised Learning in CG</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>~10:55</td>
<td>Learning on Unstructured Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>~11:35</td>
<td>Learning for Simulation/Animation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>12:05</td>
<td>Discussion</td>
<td>X</td>
<td>X</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/
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/
Recap CNN

• Convolution operators

• Pooling operators
Recipe 101: Supervised Learning in CG
Recipe 101: Supervised Learning in CG

- Obtain supervision data

\[ \{ \mathbf{x}_i, \mathbf{y}_i \}_{i=1}^{k} \quad \mathbf{x}_i \in \mathbb{R}^{m \times m}, \quad \mathbf{y}_i \in \mathbb{R}^{n \times n} \]
Recipe 101: Supervised Learning in CG

- Obtain **supervision data**

\[
\{x_i, y_i\}_{i=1:k} \quad x_i \in \mathbb{R}^{m \times m}, \quad y_i \in \mathbb{R}^{n \times n}
\]

- Setup **architecture**
  - choose non-linearity (i.e., activation)
  - optimization parameters \( \Theta = \{\theta_j\} \)
Recipe 101: Supervised Learning in CG

- Obtain **supervision data**
  \[
  \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1:k} \quad \mathbf{x}_i \in \mathbb{R}^{m \times m}, \quad \mathbf{y}_i \in \mathbb{R}^{n \times n}
  \]

- Setup **architecture**
  - choose non-linearity (i.e., activation)
  - optimization parameters \( \Theta = \{\theta_j\} \)

- Setup **loss** function
  \[
  \mathcal{L}(\Theta) := \sum_i \|f_\Theta(\mathbf{x}_i) - \mathbf{y}_i\|^2
  \]
Image Classification/Feature Extraction
Image Classification/Feature Extraction
Image Classification/Feature Extraction
Image Classification/Feature Extraction
Image Classification/Feature Extraction

\[ \mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^{f} \]
Image Classification/Feature Extraction

\[
\mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^f
\]

reduced spatial information
Image Classification/Feature Extraction

\[ \mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^f \]

-reduced spatial information
-increased global information
Image Classification/Feature Extraction

$\mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^f$

- reduced spatial information
- increased global information
- increased number of channels
Image Classification/Feature Extraction

$\mathbb{R}^{m \times m \times 3} \rightarrow \mathbb{R}^f$

- Reduced spatial information
- Increased global information
- Increased number of channels

Local features (style)
Image Classification/Feature Extraction

\[
\begin{align*}
\mathbb{R}^{m \times m \times 3} & \rightarrow \mathbb{R}^f \\
\text{reduced spatial information} & \\
\text{increased global information} & \\
\text{increased number of channels} & \\
\text{local features (style)} & \rightarrow \text{global features (content)}
\end{align*}
\]
Style Transfer Applications

[ Gatys et al. 2016, CVPR ]
Style Transfer Applications

[ Gatys et al. 2016, CVPR ]
Style Transfer Applications

[CreativeAI: Deep Learning for Computer Graphics]

Supervised Learning in CG

[Styel Transfer Applications, Gatys et al. 2016, CVPR]
Style Transfer Applications

[Sty... et al. 2016, CVPR]
Style Transfer Applications

Style

Input image

Content Representations

Convolutional Neural Network

Style Representations

Content Reconstructions

Style Reconstructions

[Gatys et al. 2016, CVPR]
Optimization Formulation (Not Learning)
Optimization Formulation (Not Learning)

\[ L_{content}(p, x, l) := \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2 \]
Optimization Formulation (Not Learning)

\[
L_{\text{content}}(p, x, l) := \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2
\]
Optimization Formulation (Not Learning)

\[ L_{content}(p, x, l) := \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2 \]
Optimization Formulation (Not Learning)

\[ \mathcal{L}_{\text{content}}(p, x, l) := \sum_{i, j} \left( F_{ij}^l - P_{ij}^l \right)^2 \]

\[ G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \]
**Optimization Formulation (Not Learning)**

Source image

\[ \mathcal{L}_{content}(p, x, l) := \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2 \]

\[ G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \]

Know

\[ \mathcal{L}_{style}(a, x) := \sum_l \sum_{ij} (G_{ij}^l - A_{ij}^l)^2 \]
Optimization Formulation (Not Learning)

$$\min_I \alpha \mathcal{L}_{content}(P_{content}, I) + \beta \mathcal{L}_{style}(A_{style}, I)$$
Optimization Formulation (Not Learning)

\[ \min_I \alpha L_{\text{content}}(P_{\text{content}}, I) + \beta L_{\text{style}}(A_{\text{style}}, I) \]
Optimization Formulation (Not Learning)

$$\min_I \alpha L_{content}(P_{content}, I) + \beta L_{style}(A_{style}, I)$$
Optimization Formulation (Not Learning)

$$\min_I \alpha L_{content}(P_{content}, I) + \beta L_{style}(A_{style}, I)$$
Optimization Formulation (Not Learning)

$$\min_{I} \alpha L_{content}(P_{content}, I) + \beta L_{style}(A_{style}, I)$$

[Deep Image Prior, Ulyanov et al. 2018, CVPR]
What We Learned?

- **CNN features**: style versus content
UNet Architecture
UNet Architecture: Image Translation
UNet Architecture: Image Translation
UNet Architecture: Image Translation

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG
UNet Architecture: Image Translation
Sketch Simplification

[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]
Sketch Simplification

[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]
Sketch Simplification

[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]
Sketch Simplification

[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]
Sketch Simplification

[Simon-Serra et al. 2016, SIGGRAPH]
[Li et al. 2017, SIGGRAPH]
Sketch Simplification: *Learning to Simplify*
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
UNet Architecture: Image Translation
UNet Architecture: Image Translation
UNet Architecture: Image Translation

UNet or ‘Hourglass’ with skip connections
Denoising Renderings

[Bako et al. 2017, SIGGRAPH]
[Chaitanya et al. 2017, SIGGRAPH]
Image Decomposition

[Narihira et al. 2015, ICCV]
[Zhou et al. 2015, ICCV]
[Innamorati et al. 2017, EGSR]
Image Decomposition: Decomposing Single Images for Layered Photo Retouching

[Innamorati et al. 2017, EGSR]
Results: Intrinsic Decomposition
Results: Intrinsic Decomposition
Results: Intrinsic Decomposition

Input

Occlusion

Albedo

Irradiance

Specular

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG
Directional Decomposition

Input

Normals

Top

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG
Directional Decomposition

Input

Normals

Bottom
Directional Decomposition

Input

Normals

Left
Directional Decomposition

Input

Normals

Right
With Inferred Layered Representation
With **Inferred** Layered Representation
3D CNN: Object Recognition

[Xiao et al. 2015, CVPR]
3D CNN: Object Recognition

[Xiao et al. 2015, CVPR]
VoxNet: Object Recognition

[Maturana et al. 2015, IROS]
VoxNet: Object Recognition

Input
(3.2 \text{ m})^3

Point Cloud

Occupancy Grid

32 \times 32 \times 32

32 \times 32 \times 32 \text{ Filters}, \text{ stride 2}

32 \times 32 \times 32 \text{ Max Pooling}

6 \times 6 \times 6 \text{ @ 32}

14 \times 14 \times 14 \text{ @ 32}

Output

128 Fully Connected

K Fully Connected

Car

[Maturana et al. 2015, IROS]
Multi-view CNN for 3D

3D shape model rendered with different virtual cameras

[Su et al. 2015, ICCV]
Multi-view CNN for 3D

Su et al. 2015, ICCV

3D shape model rendered with different virtual cameras

2D rendered images
Multi-view CNN for 3D

[Su et al. 2015, ICCV]
Multi-view CNN for 3D

[Su et al. 2015, ICCV]
Mesh Labeling / Segmentation

(a) candelabra  (b) lamp  (c) guitar

(d) armadillo  (e) chair  (f) fourleg

(g) ant  (h) human  (i) fish

[Guo et al. 2016, ACM TOG]
Audio-driven Facial Animation

[Karras et al. 2017, SIGGRAPH]
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
Colorization

• *Let there be Color!,* Iizuka et al., 2016
• *Colorful Image Colorization,* Zhang et al. 2016
• *Learning Representations for Automatic Colorization,* Larsson et al., 2016
• *Real-Time User-Guided Image Colorization with Learned Deep Priors,* Zhang et al. 2017
Colorization: *Let There Be Color!*
Colorization: *Let There Be Color!*
UNet + Global Features
UNet + Global Features
UNet + Global Features
Single-image SVBRDF Capture [Deschaintre et al. 2018, Siggraph]
UNet with Global Features
Importance of Global Features

<table>
<thead>
<tr>
<th>Specular albedo</th>
<th>Roughness</th>
<th>Diffuse albedo</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>ground truth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNet</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG
Importance of Global Features

<table>
<thead>
<tr>
<th>Specular albedo</th>
<th>Roughness</th>
<th>Diffuse albedo</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>ground truth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unet + global f.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Realistic Reconstructions

Input (Wood)

Real pictures

Ours
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
Rendering Loss: Render Function inside the Network

Input → CNN → Prediction → Renderer → Rendered Image

Ground truth → Rendered Image → Loss
Rendering Loss: Render Function inside the Network

comparing parameter values versus effect of the parameters
UNet with Global Features
Extension to Multiple Images

[Deschaintre et al. 2019, EGSR]
Extension to Multiple Images

[Deschaintre et al. 2019, EGSR]
Result

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Renderings</th>
<th>Normal</th>
<th>Diffuse albedo</th>
<th>Roughness</th>
<th>Specular albedo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 input</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
<tr>
<td>2 inputs</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
<tr>
<td>3 inputs</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Self-supervision: Regression

[Li et al. 2017, Siggraph]
Self-supervision: Regression

$$\{ x_i, y_i \}$$
Self-supervision: Regression

\[ \{x_i, y_i\} \]

\[ y = f(x) \]

[Li et al. 2017, Siggraph]
Self-supervision: Regression

\[ \{x_i, y_i\} \]

[Li et al. 2017, Siggraph]

\[ y = f(x) \]
Self-supervision: Regression

\[ \{ x_i, y_i \} \]

\[ y = f(x) \]

[Li et al. 2017, Siggraph]
Self-supervision: Regression

\[ \{x_i, y_i\} \]

\[ y = f(x) \]

[Li et al. 2017, Siggraph]
Self-supervision: Regression

\[ \{x_i, y_i \} \]

\[ y = f(x) \]

[Li et al. 2017, Siggraph]
Self-supervision: Regression

[Li et al. 2018, Siggraph]

Input without SVBRDF label
Self-supervision: Regression

[Li et al. 2018, Siggraph]
Self-supervision: Regression

[Li et al. 2018, Siggraph]
Self-supervision: Regression

[Li et al. 2018, Siggraph]
Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]
Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]

• Look back at image formation model (rendering equation)
Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]

• Look back at image formation model (rendering equation)

3D model
Differentiable Rendering: Rendering in the Loop

- Look back at image formation model (rendering equation)

view sample → 3D model → color, material, Illumination → image-formation model

[Henzler et al. 2019, ICCV]
Differentiable Rendering: Rendering in the Loop

• Look back at image formation model (rendering equation)

[Henzler et al. 2019, ICCV]
Differentiable Rendering: Rendering in the Loop

[Henzler et al. 2019, ICCV]

• Look back at image formation model (rendering equation)

• Image formation, view transformation are known functions/transformations

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG
UNet Revisited

\[ F \]  

\( F \) is a differential but known (CG) function (e.g., rendering, camera matrix, simulation)
UNet Revisited

$F$ differential but known (CG) function (e.g., rendering, camera matrix, simulation)
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
- **CG-specific functions**: custom blocks embedded into networks (e.g., camera model, differentiable rendering)
Encoder Decoder

\[ x \xrightarrow{E} z \xrightarrow{D} D(z) \]
Encoder Decoder

\[ \mathcal{L}_\Theta := \sum_i \| D(E(x_i) - y_i) \|^2 \]

encoder-decoder
Encoder Decoder

Encoder-Decoder

\[ \mathcal{L}_\Theta := \sum_i \| D(E(x_i)) - y_i \|^2 \]

Autoencoder

\[ \mathcal{L}_\Theta := \sum_i \| D(E(x_i)) - x_i \|^2 \]
Design Options

1. sketching

[Wang et al. 2018, Siggraph Asia]
Design Options

1. sketching
2. sewing patterns

[Wang et al. 2018, Siggraph Asia]
Design Options

1. sketching
2. sewing patterns
3. draped garment

[Wang et al. 2018, Siggraph Asia]
Design Options

1. sketching
2. sewing patterns
3. draped garment

= interaction(sewing pattern, material, body shape)

[Wang et al. 2018, Siggraph Asia]
Interaction through Simulation
Interaction through Simulation

*realistic simulations but NOT interactive*
Learning a Latent Space (AutoEncoder)

input sketch ($\mathbf{S}$) → DenseNet → sketch descriptor ($\mathbf{S}$) → PCA coeff. ($\mathbf{M}$) → draped garment ($\tilde{\mathbf{M}}$)
Learning a Latent Space (AutoEncoder)
Learning a Latent Space (AutoEncoder)
Learning a Latent Space (AutoEncoder)
Learning a **Shared** Latent Space (3-way AutoEncoder)

- **input sketch (S)**
- **sketch descriptor (S)**
- **PCA coeff. (M)**
- **draped garment (\(\tilde{M}\))**
- **body shape parameters (B)**
- **garment parameters (G)**

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG
Learning a **Shared** Latent Space (3-way AutoEncoder)

- **input sketch** (S)
- **sketch descriptor** (S)
- **PCA coeff.** (M) **draped garment** (\(\tilde{M}\))
- **body shape parameters** (B)
- **garment parameters** (G)

**CreativeAI: Deep Learning for Computer Graphics**  **Supervised Learning in CG**
Learning a **Shared** Latent Space (3-way AutoEncoder)

- **Input sketch (S)**
- **DenseNet**
- **Sketch descriptor (S)**
- **Latent space**
- **PCA coeff. (M) draped garment (\(\tilde{M}\))**
- **Body shape parameters (B)**
- **Garment parameters (G)**

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG
Sketch editing:

[Image of a sketch editing software interface]
Network as a Learned ‘Basis’
Network as a **Learned ‘Basis’**
Network as a Learned ‘Basis’
Network as a Learned ‘Basis’
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
- **Autoencoder**: category-specific non-linear basis
Conditional Decoder

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG
Conditional Decoder

\[ z \]

\[ D \]

\[ D(z, z') \]
Network for Compression

[Rainer et al. 2019, Eurographics]
Network for Compression

[Original BTF]

[PCA]

[Rainer et al. 2019, Eurographics]
Network for Compression

[Original BTF] [Neural BTF] [PCA]

[Rainer et al. 2019, Eurographics]
Learning Volumetric Deformation [Yumner et al. 2016, ECCV]
Learning Volumetric Deformation

CreativeAI: Deep Learning for Computer Graphics  Supervised Learning in CG

[Yumner et al. 2016, ECCV]
Network for (BTF) Compression
Network for (BTF) Compression
Sequence Prediction (past matters)

\[ a_1=2 \quad a_2=0 \quad a_3=1 \quad a_4=3 \quad a_5=4 \quad a_6=2 \quad a_7=5 \]

\[ x = \text{bringen sie bitte das auto zurück} \]

\[ y = \text{please return the car} \]
Neural Nets

\[ X_i \rightarrow \text{NN} \rightarrow Y_i \]
Neural Nets

\[ y_i \leftarrow f_{\Theta}(x_i) \]

CreativeAI: Deep Learning for Computer Graphics
Recurrent Neural Nets

\[
X_i \xrightarrow{\text{NN}} Y_{i+1}
\]

CreativeAI: Deep Learning for Computer Graphics
Recurrent Neural Nets

\[ y_{i+1} \leftarrow f_\Theta(x_i, y_i) \]
Recurrent Neural Nets

\[ y_{i+1} \leftarrow f_\Theta(x_i, y_i) \]

LSTM
GRU (popular in CG)
Single Image Facial Relighting

[Sun et al. 2019, Siggraph]
Single Image Facial Relighting

[Sun et al. 2019, Siggraph]
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
- **Autoencoder**: category-specific non-linear basis
- **Conditional decoder**: auxiliary input (e.g., user control, environmental variables)
Network as a Parameterization

Bootstrapping

Single top view measurement
Single image initialization
SVBRDFs
Network as a Parameterization

\[ \arg \min_s \sum_i \mathcal{L}(I_i, R(s, C_i)) \]
Network as a **Parameterization**

\[ \arg \min_s \sum \mathcal{L}(I_i, R(s, C_i)) \]

\[ \arg \min_s \sum \mathcal{L}(I_i, R(D(z), C_i)) \]
Network as a Parameterization

\[
\arg\min_s \sum_i \mathcal{L}(I_i, R(s, C_i))
\]

\[
\dim(z) \leftarrow \dim(s)
\]

\[
\arg\min_s \sum_i \mathcal{L}(I_i, R(D(z), C_i))
\]
Network as a Parameterization

$$\text{arg min}_s \sum_i \mathcal{L}(I_i, R(s, C_i))$$

$$\text{dim}(z) \ll \text{dim}(s)$$

$$\text{arg min}_s \sum_i \mathcal{L}(I_i, R(D(z), C_i))$$
What We Learned?

- **CNN features**: style versus content
- **UNet**: for (image) translation problems
- **UNet + Skip connection**: preserves details
- **UNet + Skip + global features**: access to global/non-local information
- **Conditional decoder**: auxiliary input (e.g., user control, environmental variables)
- **Autoencoder**: category-specific non-linear basis
- **Sequences**: RNN, LSTM, GRU (not covered in this course)
- **CG-specific functions**: custom blocks embedded into networks (e.g., camera model, differentiable rendering)
- **Learned regularizer**: Optimize over learned network (e.g., decoder)
Course Information (slides/code/comments)

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