Material Editing Using a Physically Based Rendering Network

Hanxiao He
Feb 14th, 2018
Motivation: Material Editing

- **Change the Material of Objects**

- **Applications**
  - **Image Editing**: generating new image with new material without 3D information
  - **Enable New Workflow**: pick material from a reference image
  - **Fast Preview in 3D Design**: avoid expensive re-rendering
Challenges

Ill-defined Checker shadow illusion

Target 2D Image

Infer

Shape
Material
Illumination

Intrinsic Properties

Edit

Shape'
Material'
Illumination'

Output Image

Intrinsic Properties

Shape A
Material A
Illumination A

Shape B
Material B
Illumination B

III defined Checker shadow illusion
Multiple combinations of intrinsic properties may result in the same image.
Architecture: overview
Architecture: encoder
Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture
Architecture: feature representation
Material Prediction

• 108 material coefficients $\mathbf{m}$

Normal Prediction

• For every pixel $p$ in input image, we will have its corresponding normal $\mathbf{n}_p$
• The shape of $\mathbf{n}$ is $(3 \times \text{input width} \times \text{input height})$

Illumination Prediction

• The pixel coordinate is mapped to spherical coordinates and corresponds to an incoming light direction $\mathbf{w}_i$
• The pixel value indicates the intensity of the light
• The shape of $\mathbf{L}$ is $(3 \times 64 \times 128)$
Architecture: rendering layer (natural decoder)
BRDF (Bidirectional reflectance distribution function), which defines how the light with incoming direction $\omega_i$ is reflected along the outgoing direction $\omega_o$.

- $f(\omega_i, \omega_o, m)$: BRDF (Bidirectional reflectance distribution function), which defines how the light with incoming direction $\omega_i$ is reflected along the outgoing direction $\omega_o$.
- $L(\omega_i)$: The intensity of the light with incoming direction $\omega_i$.
- $I_p(n_p, m, L) = \sum_I f(\omega_i, \omega_o, m)L(\omega_i)\max(0, n_p \cdot \omega_i)d\omega_i$.
- $\omega_i$: direction of incoming light.
- $\omega_o$: direction of outgoing light.
- $n_p$: surface normal information.
- $m$: material coefficient.
- $L$: illumination information.
- $I_p$: outgoing light intensity for every pixel $p$ in input image $I$. 

Target material ($m_t$), input image ($I$), and synthesized target image ($O'$) are shown in the diagram.
\[ I_p(n_p, m, L) = \sum_{L} f(\omega_i, \omega_o, m) L(\omega_i) \max(0, n_p \cdot \omega_i) \, d\omega_i \]

- \( f(\omega_i, \omega_o, m) \): BRDF (Bidirectional reflectance distribution function), which defines how the light with incoming direction \( \omega_i \) is reflected along the outgoing direction \( \omega_o \).
- \( L(\omega_i) \): The intensity of the light with incoming direction \( \omega_i \).

**BRDF**

- **Traditional BRDF**: no parameters, and is in the form of lookup table.
- **Directional Statistic BRDF (DSBRDF)**: represents each BRDF as a combination of hemispherical exponential power distributions. The number of the parameters depends on the number of distributions used (here is 108).

- \( \omega_i \): direction of incoming light
- \( \omega_o \): direction of outgoing light
- \( n_p \): surface normal information
- \( m \): material coefficient
- \( L \): illumination information
- \( I_p \): outgoing light intensity for every pixel \( p \) in input image \( I \)
Architecture: refinement (optional)

- **Reason**
  - Training data are synthetic, variation difference do exist

- **Method (post-optimization)**

\[
\arg\min_{n^*, m^*, L^*} \| I^* - I \|^2 + a \| n^* - n' \|^2 + b \| L^* - L' \|^2
\]

- \( I^* \) is the image formed by \( n^*, m^*, L^* \)
- \( n', m', L' \) is the prediction of the network
- \( n^*, m^*, L^* \) is the optimized for surface normal, material and illumination
- **LBFGS** (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) to solve the argmin equation (unconstrained nonlinear optimization problem)
Training: loss functions
• **Normal**: L2 loss ($l_{normal}$)
  - $l_{normal} = \sum_p (n_p - n'_p)^2$

• **Material**: L2 loss ($l_{material}$)
  - $l_{material} = (m - m')^2$

• **Image**: perceptual loss ($l_{perceptual}$)

• **Final Combination** ($l$)
  - $l = w_n l_{normal} + w_m l_{material} + w_p l_{perceptual}$

Perceptual losses for real-time style transfer and super-resolution
\[ I_p + 5 = I'_p \]

Naïve loss = \((5)^2 \times \text{width} \times \text{height}\)
Training: procedure
• Material prediction and normal prediction module are pre-trained using L2 loss for a few iterations
• Train the whole network jointly
• Some hyper-parameters are changed (momentum parameters, learning rate)
Evaluation: datasets

- **Synthetic source**
  - 280 3D models (130 cars, 50 chairs, 50 sofas and 50 monitors)
  - 80 different materials
  - 10 free HDR environment illumination map (random rotation to augment)

- **Generation**
  - Each 3D model, material, illumination lead to 5 synthetic image with random viewpoint

- **Size**
  - 240K pre-training
  - 480K joint category finetuning
Evaluation: metrics

- L2 (↓)
- Cosine (↓)
  - $\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$
- SSIM (structural similarity index) (↑)
  - Given two input image $x$ and $y$, the structural similarity is measured as:
    \[
    \text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}
    \]
  - where, $\mu_x$ and $\mu_y$ is the mean, $\sigma_x^2$ and $\sigma_y^2$ is the variance, $\sigma_{xy}$ is the covariance
Evaluation: rendering layer (material coefficients)
Evaluation: rendering layer (material coefficients)

- **Accuracy**

<table>
<thead>
<tr>
<th></th>
<th>material</th>
<th>rendering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L2</td>
<td>L2</td>
</tr>
<tr>
<td>standalone</td>
<td>4.1038</td>
<td>544.1</td>
</tr>
<tr>
<td>combined</td>
<td>4.8144</td>
<td>355.5</td>
</tr>
</tbody>
</table>

L2 metric for material may not be sufficient

- **Synthesized image**
Evaluation: rendering layer (surface normals)
Evaluation: rendering layer (surface normals)

- **Accuracy**

<table>
<thead>
<tr>
<th></th>
<th>normals</th>
<th>rendering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cosine</td>
<td>L2</td>
</tr>
<tr>
<td>standalone</td>
<td>0.9105</td>
<td>313.2058</td>
</tr>
<tr>
<td>combined</td>
<td>0.9050</td>
<td>238.4619</td>
</tr>
</tbody>
</table>

- **Normal prediction**
Evaluation: multi-material examples

- Accuracy
  - L2: 1272.1
  - SSIM: 0.9362
- Synthesized image
### Evaluation: comparison

<table>
<thead>
<tr>
<th>Lombardi et al.</th>
<th>Baseline 1</th>
<th>Baseline 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use priors to optimize</td>
<td>Based on autoencoder</td>
<td>Train the prediction models separately.</td>
</tr>
<tr>
<td>Provide g.t. surface normals</td>
<td>Encoder maps image to material properties and other factors</td>
<td>Render with the infer results</td>
</tr>
<tr>
<td>Render with target material, g.t. normals and predicted illumination</td>
<td>Decoder uses new material properties and original other factor to synthesize image</td>
<td></td>
</tr>
</tbody>
</table>
Evaluation: comparison (synthetic image)

- **Accuracy**

<table>
<thead>
<tr>
<th></th>
<th>Lombardi</th>
<th>Baseline1</th>
<th>Baseline2</th>
<th>ours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L2</td>
<td>no opt</td>
<td>opt</td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>802.7</td>
<td>1076.7</td>
<td>886.9</td>
<td>805.2</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9416</td>
<td>0.9173</td>
<td>0.9256</td>
<td>0.9408</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.9159</td>
<td>0.9557</td>
</tr>
</tbody>
</table>

- **Synthesized image**
Evaluation: comparison (real image)

- Synthesized image
Contributions

- Present a rendering layer to enable end-to-end network training
  - Rendering layers help to obtain more detailed normal prediction and better material prediction.
  - Really?

- Achieve better results on material editing tasks.
Evaluation: rendering layer (material coefficients)

- **Accuracy**
  
<table>
<thead>
<tr>
<th>material</th>
<th>rendering</th>
<th>L2</th>
<th>L2</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>standalone</td>
<td>4.1038</td>
<td>544.1</td>
<td>0.9297</td>
<td></td>
</tr>
<tr>
<td>combined</td>
<td>4.8144</td>
<td>355.5</td>
<td>0.9517</td>
<td></td>
</tr>
</tbody>
</table>

L2 metric for material may not be sufficient

- **Synthesized image**
Future work

- Design a robust loss function for illumination model
- Let material model to perform per-pixel material predictions
- Model the rendering layer function more accurately to take incorporation of advanced light interactions into consideration (crucial for real images and multiple objects)
Thanks