Texture attribute synthesis and transfer using feed-forward CNNs

Thomas Irmer, Tobias Glasmachers, Subhransu Maji

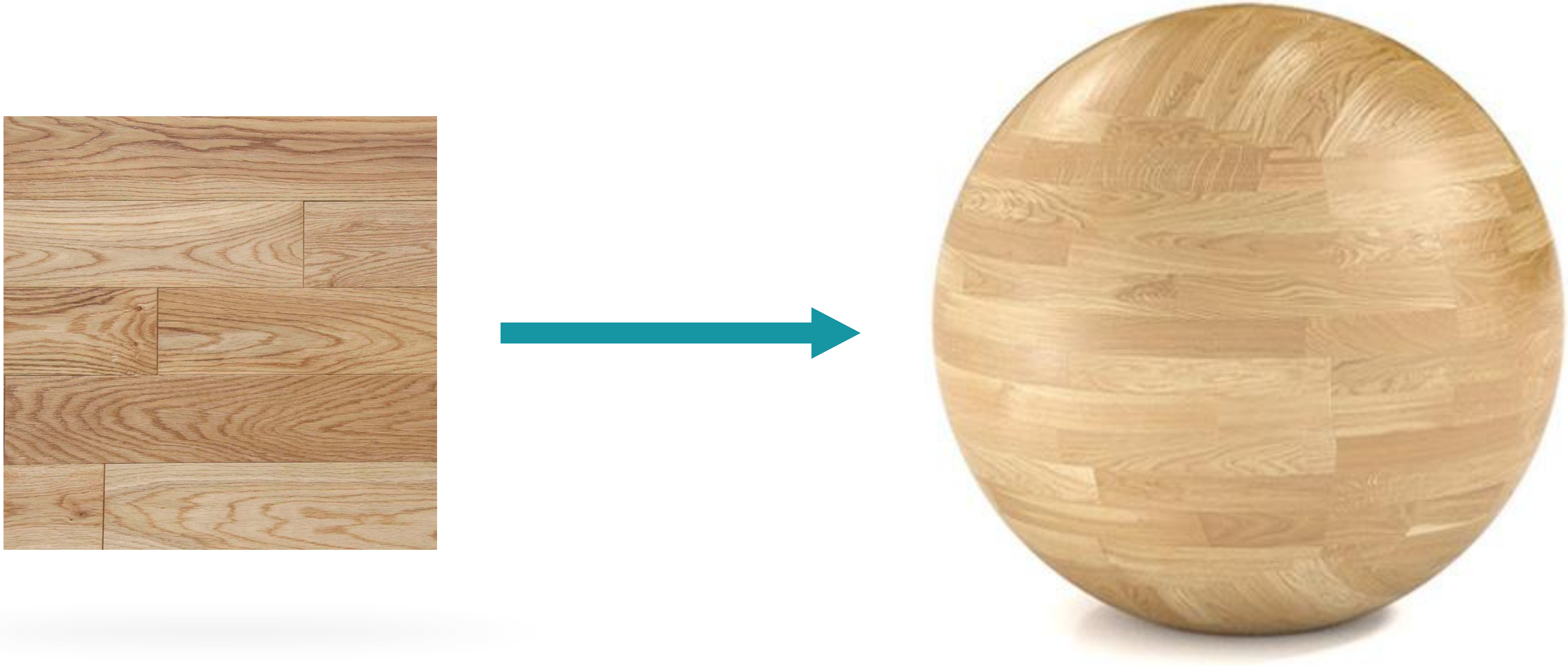
RUHR UNIVERSITÄT BOCHUM RUB

UMass Amherst

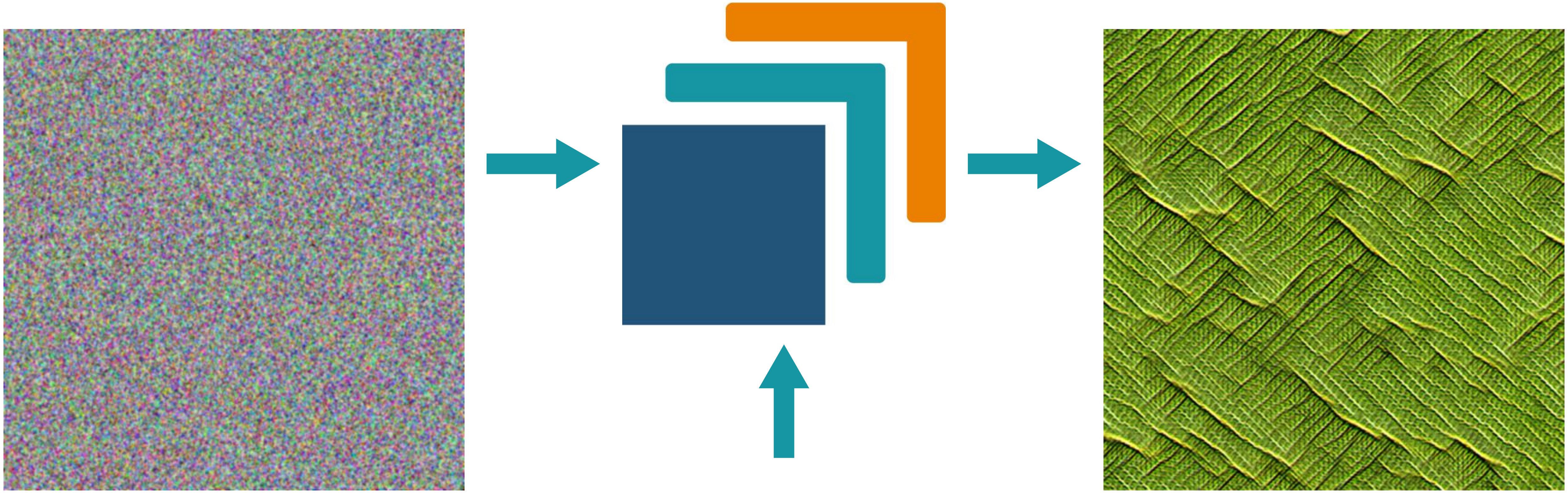
Paper-ID 175

Texture attribute synthesis and transfer using feed-forward CNNs

Thomas Irmer, Tobias Glasmachers, Subhransu Maji

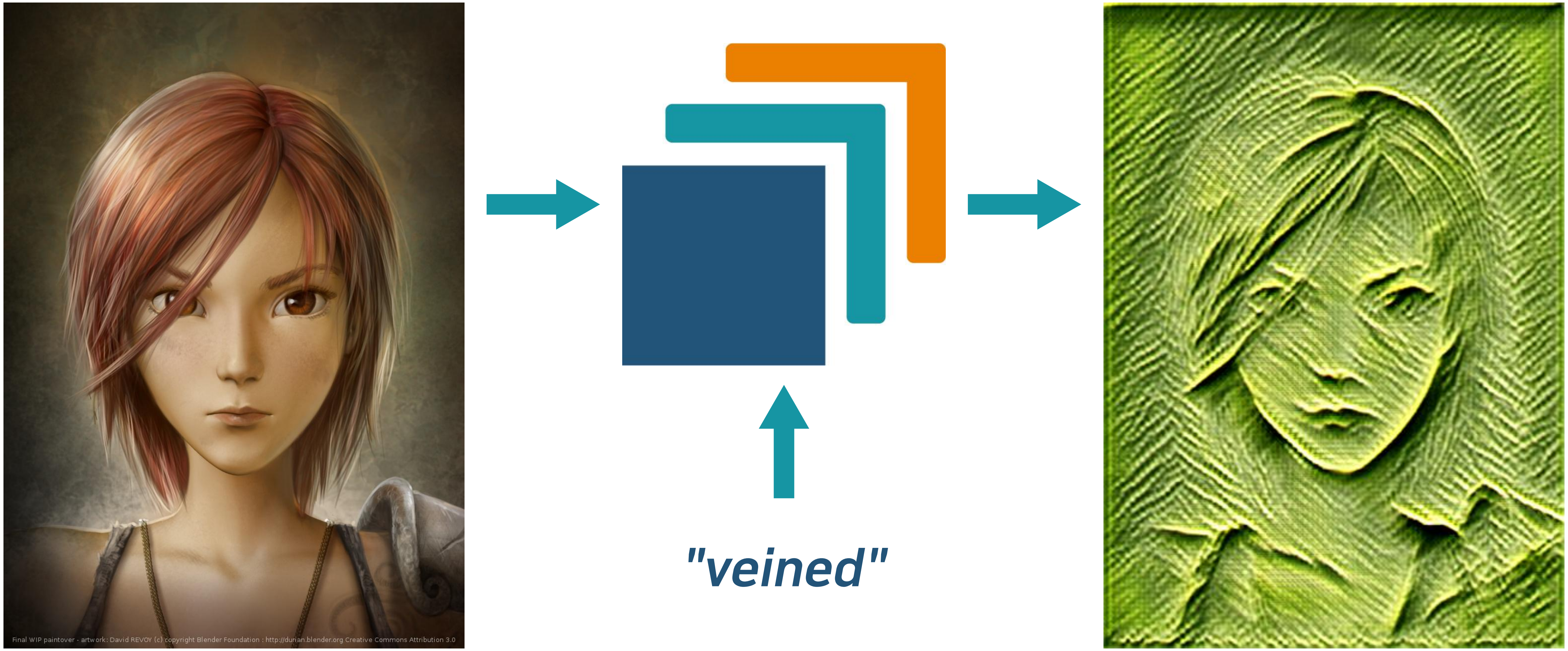


Texture attribute synthesis



"veined"

Texture attribute transfer



"veined"

Final WIP paintover - artwork: David REVOY (c) copyright Blender Foundation : http://durian.blender.org Creative Commons Attribution 3.0

Method

- **Apply** *texture features*.
- **Preserve** *content features*.

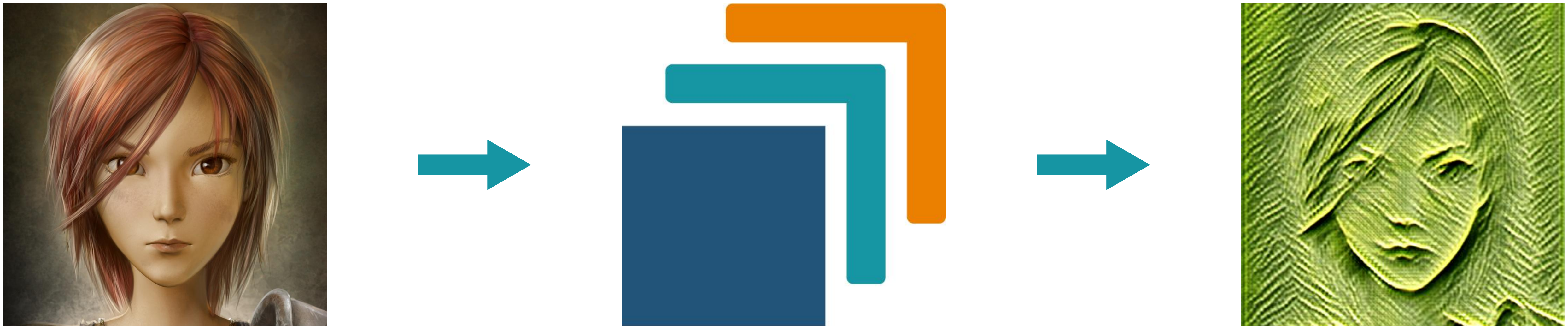
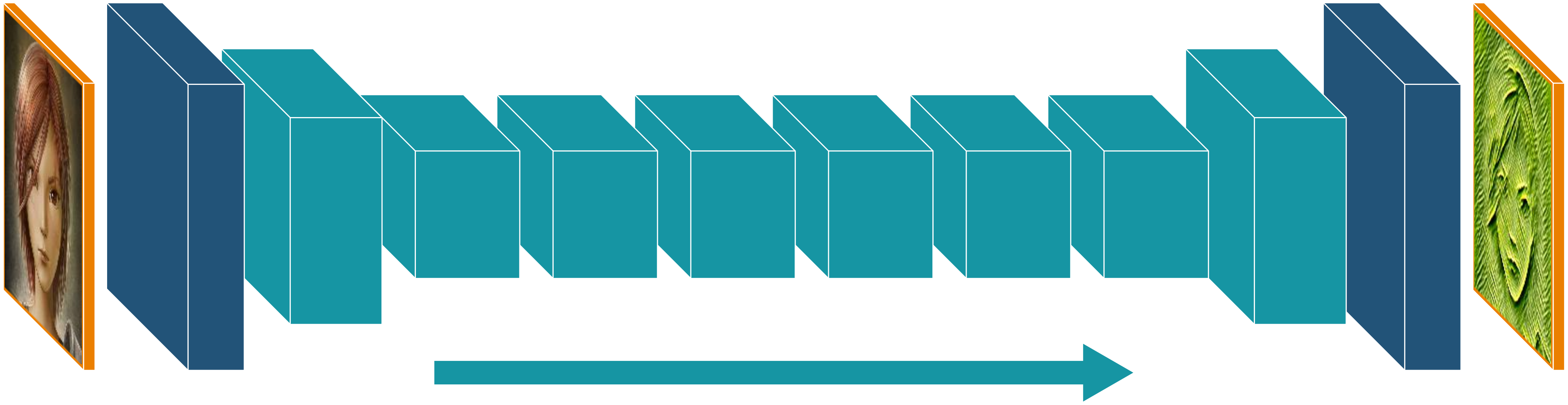


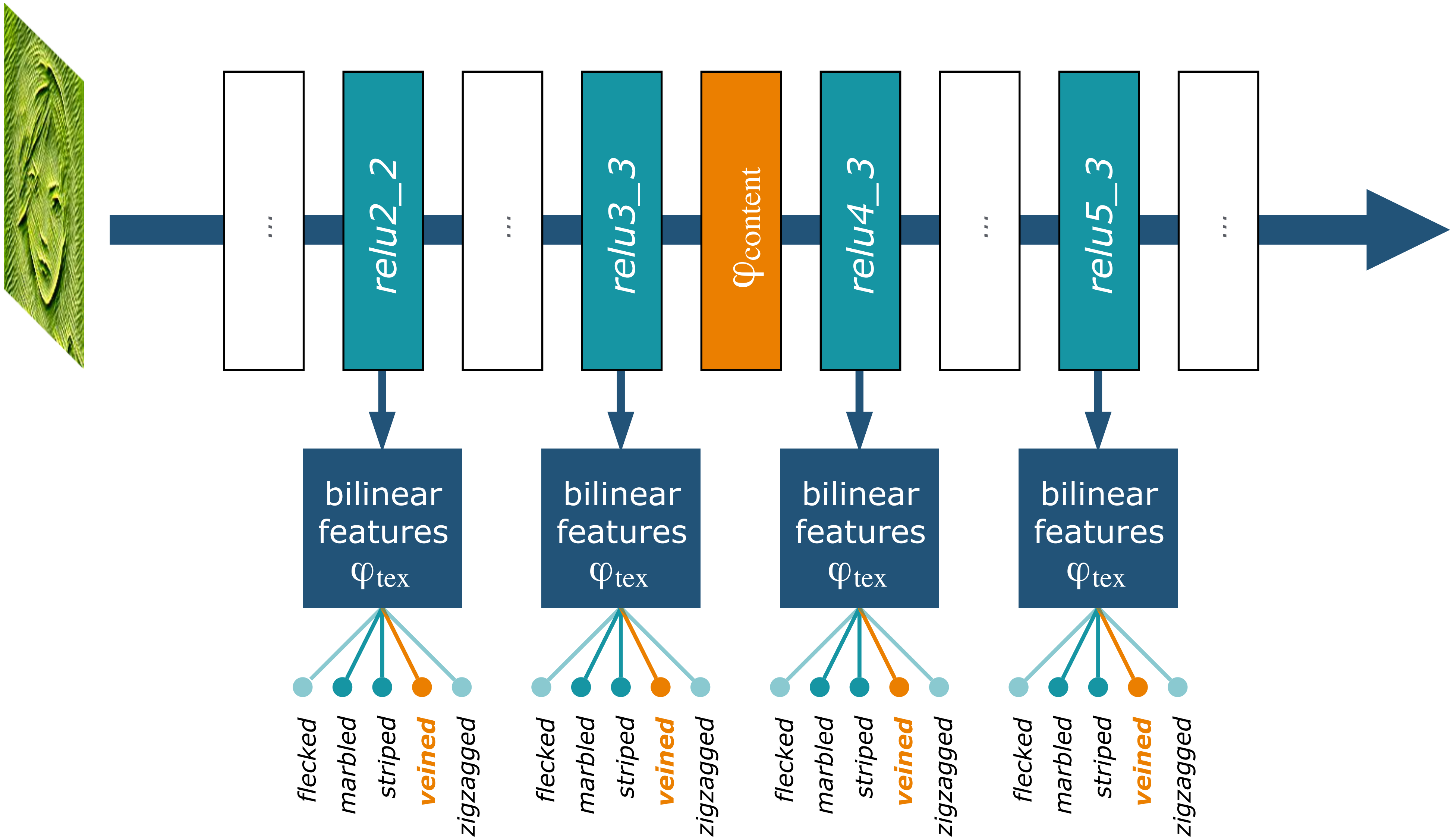
Image transform network f_t

- **Applies** *texture features* through learned filters.

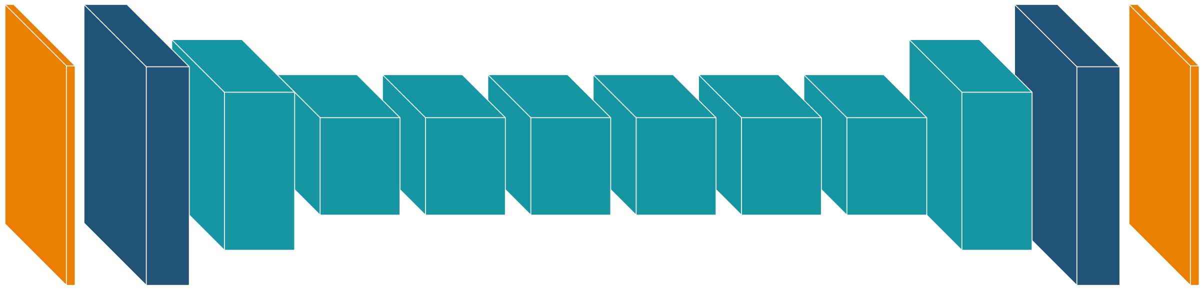


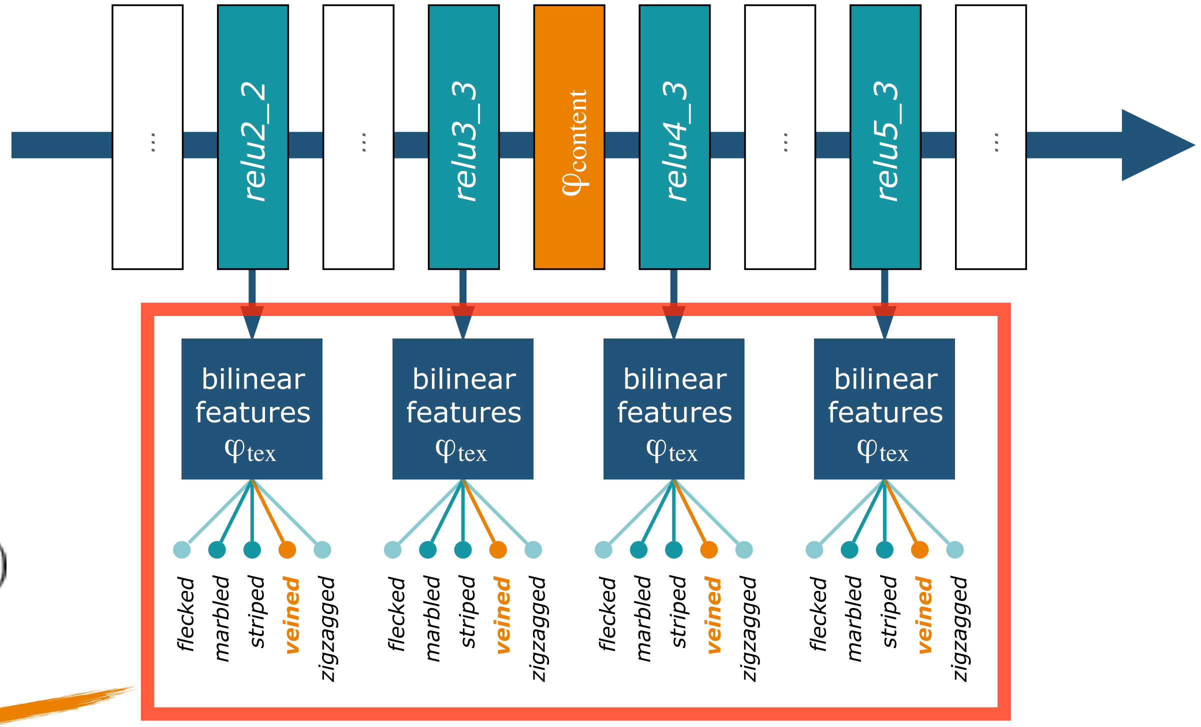
(residual) convolutions / deconvolution

Texture target network θ



Loss function





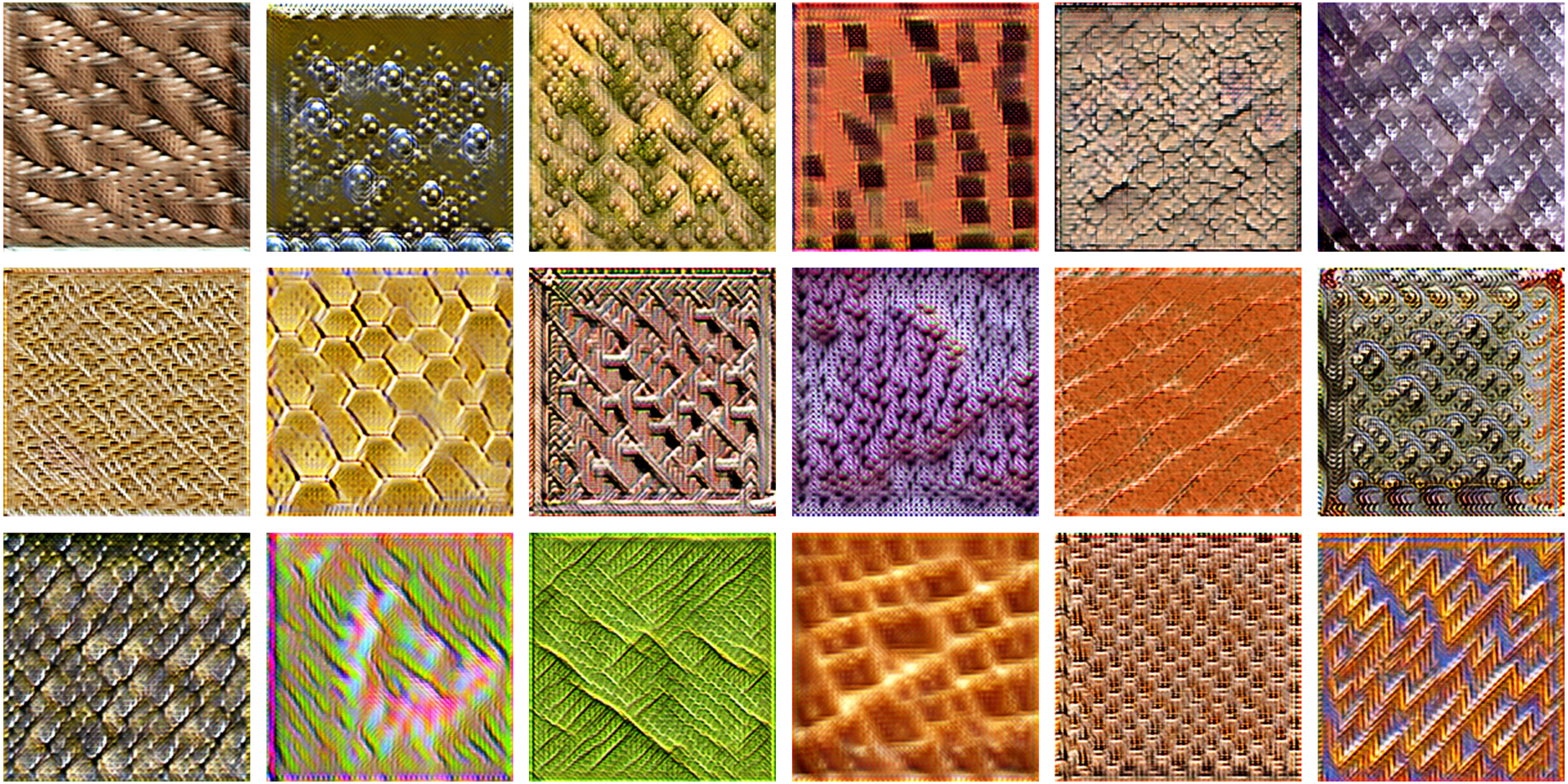
$$\mathcal{L}_{\text{total}}(\vec{x},\hat{\vec{x}}) = \sum_{i=1}^{n} \boxed{\mathcal{L}^{(i)}_{\text{texture}}(t,\hat{\vec{x}})} + \mathcal{L}_{\text{content}}(\vec{x},\hat{\vec{x}}) + \mathcal{L}_{\text{prior}}(\hat{\vec{x}})$$

$$\mathcal{L}^{(l)}_{\text{texture}}(t,\hat{\vec{x}}) = \left(\phi^{(l)}_{\text{texture}}(t) - \phi^{(l)}_{\text{texture}}(\hat{\vec{x}})\right)^2$$

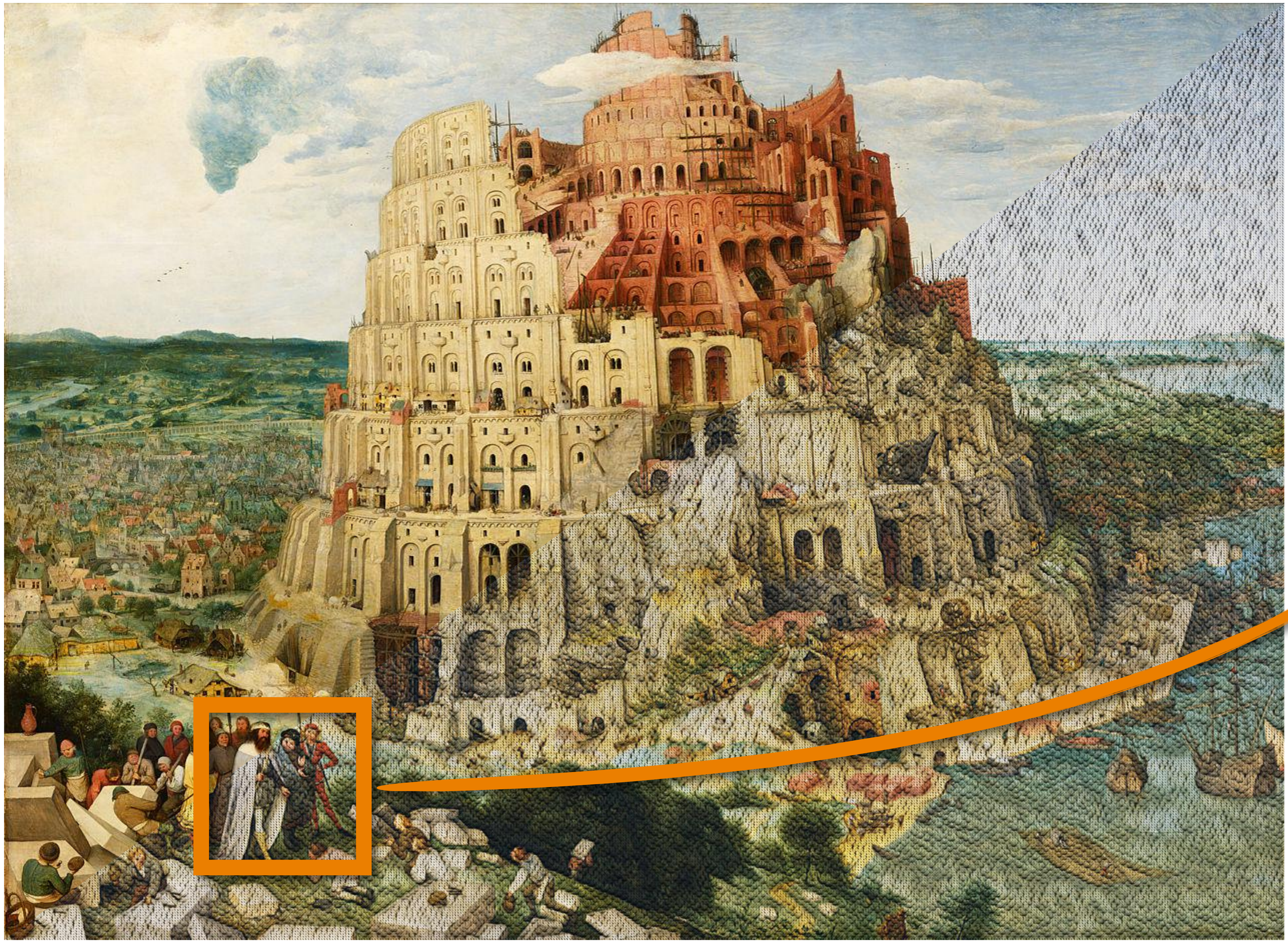
$$\mathcal{L}_{\text{content}}(\vec{x},\hat{\vec{x}}) = \left(\phi_{\text{content}}(\vec{x}) - \phi_{\text{content}}(\hat{\vec{x}})\right)^2$$

$$\mathcal{L}_{\text{prior}}(\hat{\vec{x}}) = \sum_{i,j}\left(\left(\hat{\vec{x}}_{i,j+1} - \hat{\vec{x}}_{ij}\right)^2 + \left(\hat{\vec{x}}_{i+1,j} - \hat{\vec{x}}_{ij}\right)^2\right)^{\beta/2}$$

Results - Texture Synthesis



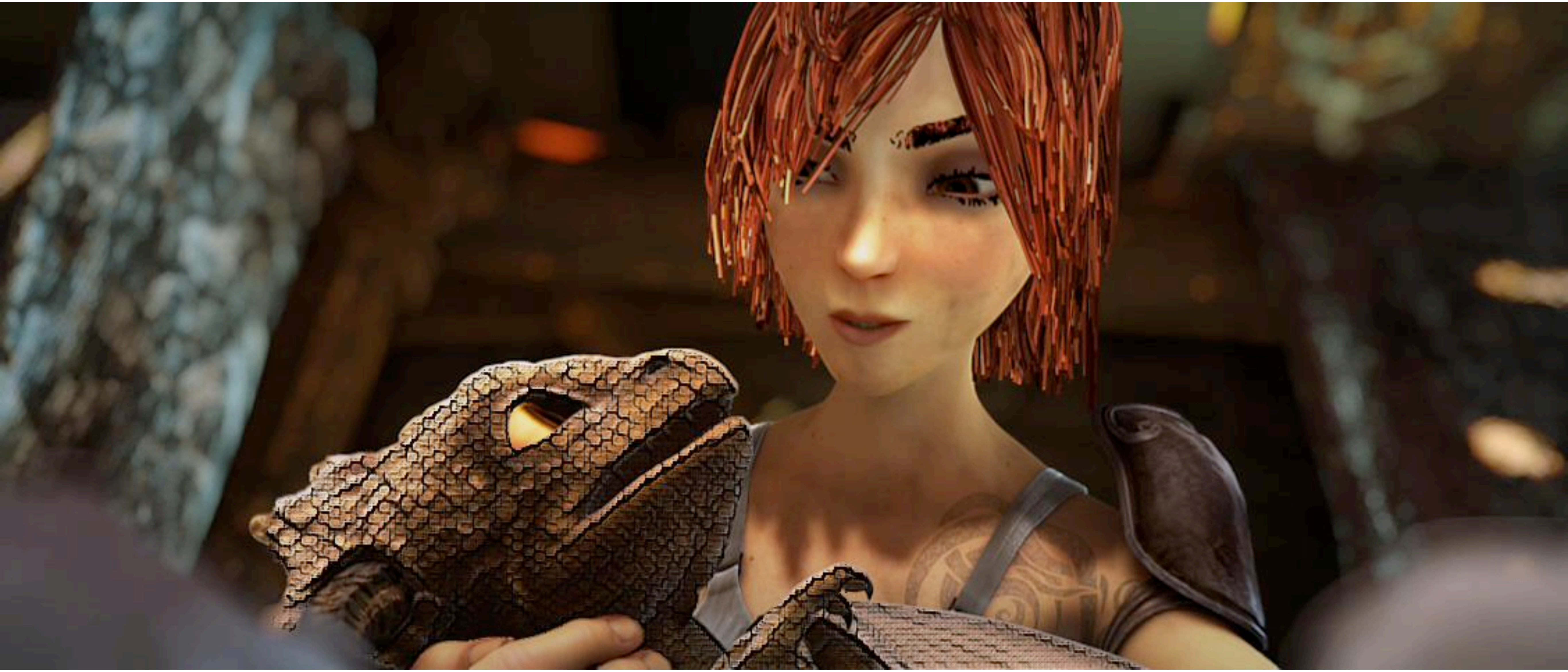
Results - Texture Transfer



"knitted"

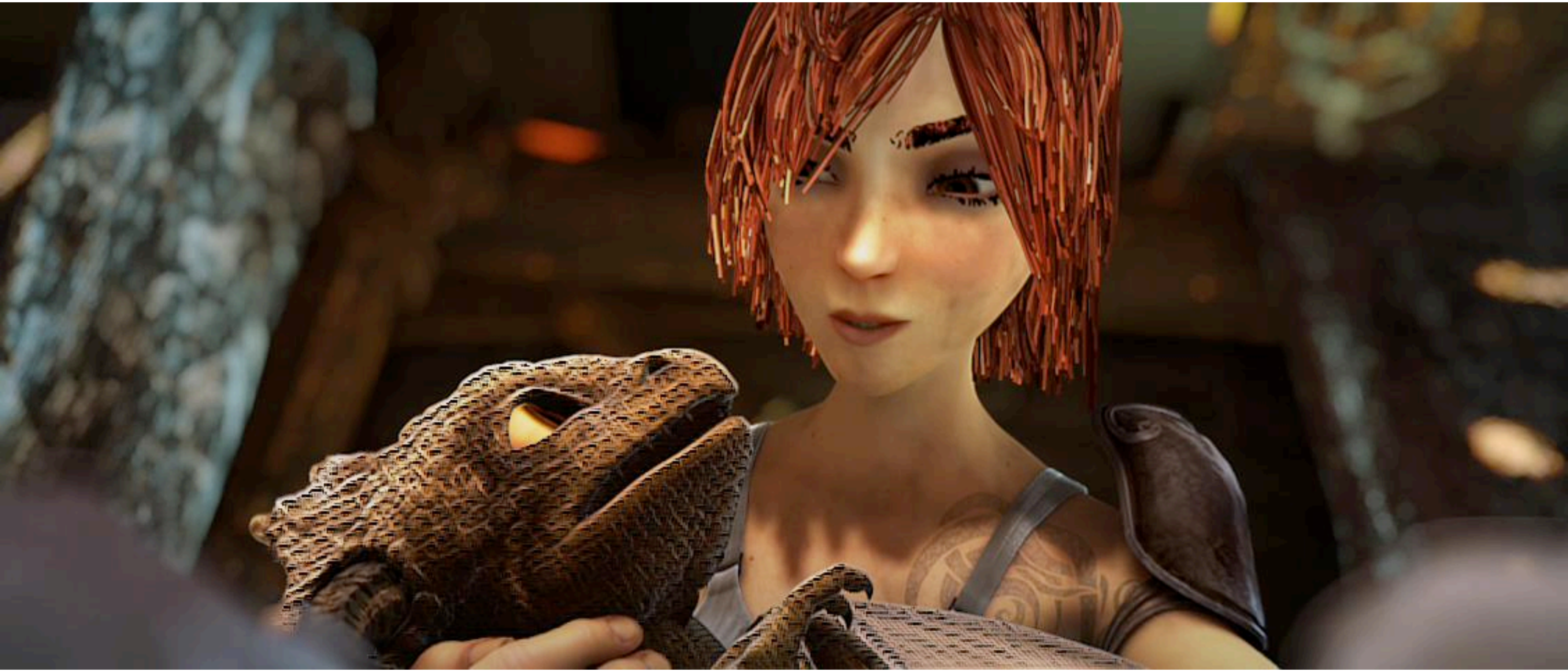
1920 x 1440 pixels

Results - Texture Transfer



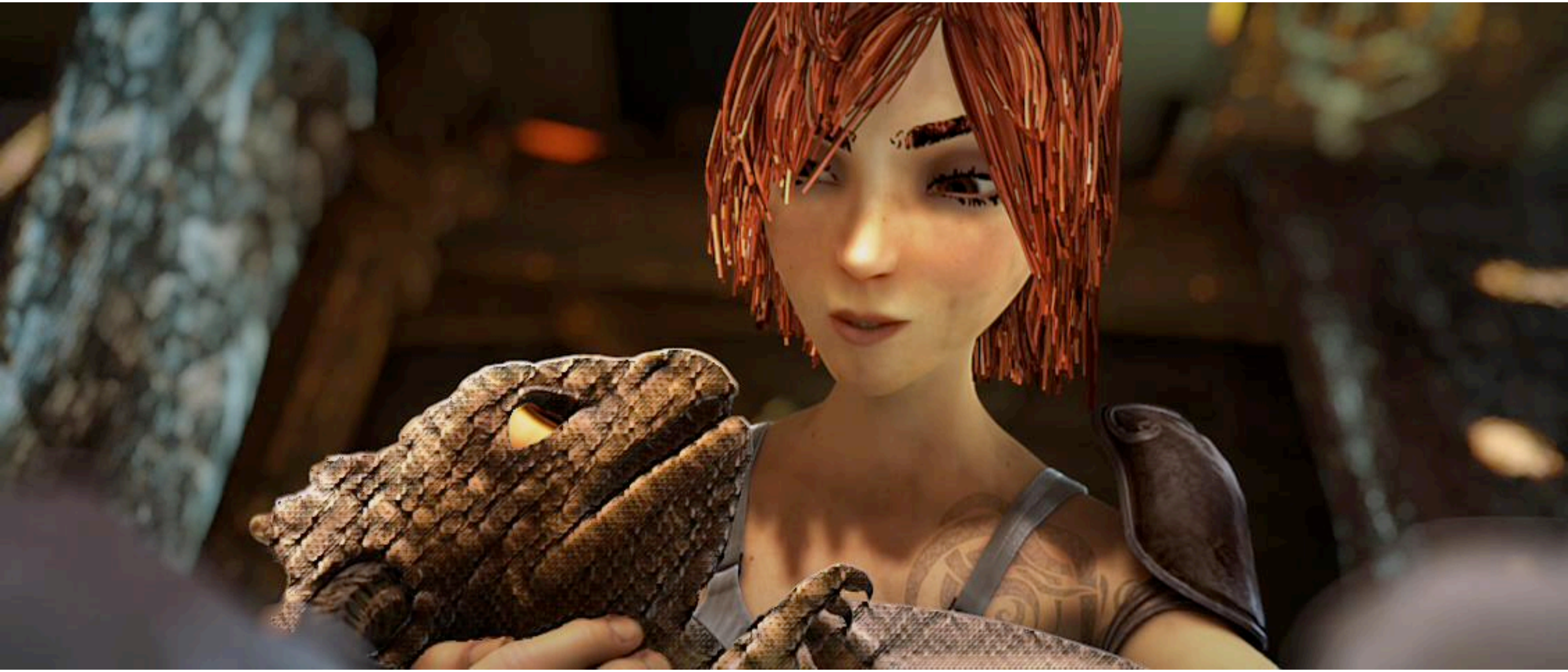
"cracked"

Results - Texture Transfer



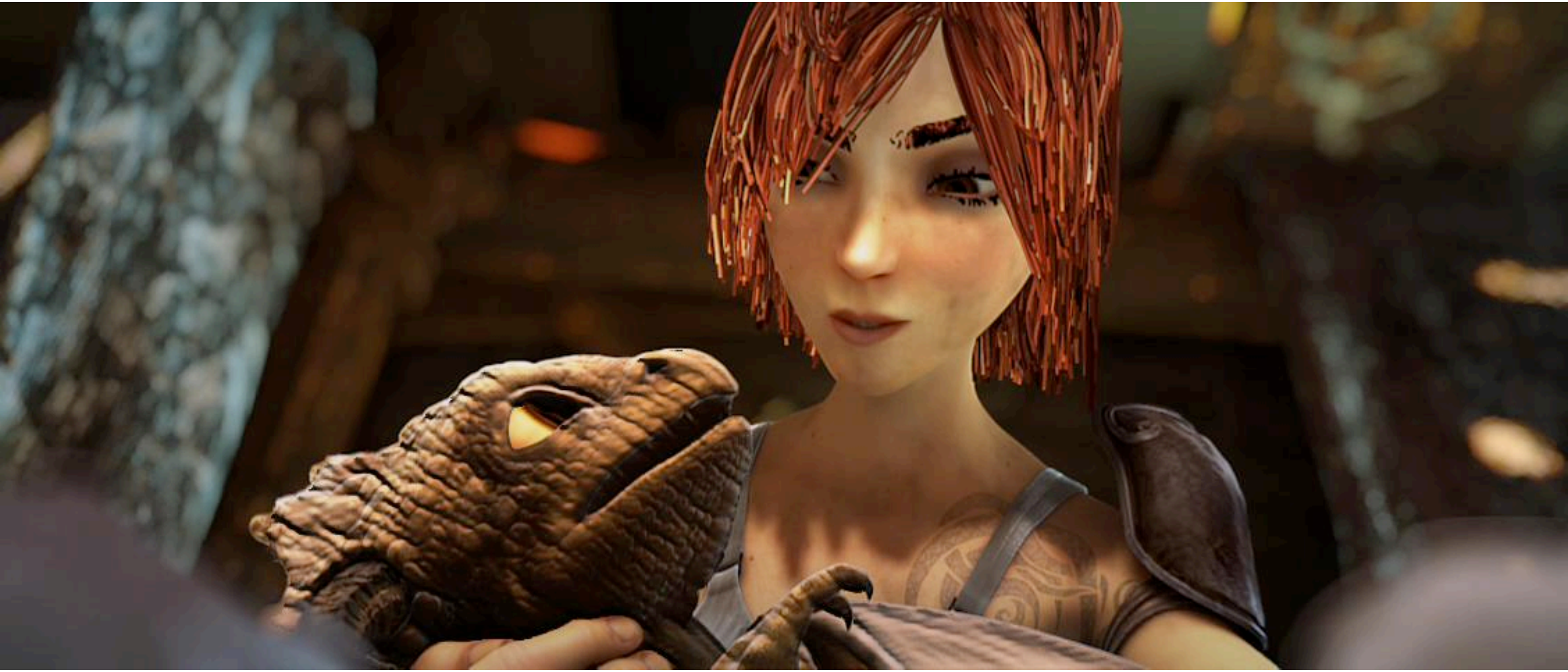
"fibrous"

Results - Texture Transfer



"scaly"

Results - Texture Transfer



"wrinkled"

Thank you!