Texture attribute synthesis and transfer using feed-forward CNNs

Thomas Irmer, Tobias Glasmachers, Subhransu Maji
Texture attribute synthesis and transfer using feed-forward CNNs
Thomas Irmer, Tobias Glasmachers, Subhransu Maji
Texture attribute synthesis

"veined"
Texture attribute transfer

"veined"
Method

- **Apply** *texture features*.
- **Preserve** *content features*.
Image transform network $f_t$

- Applies *texture features* through learned filters.

(residual) convolutions / deconvolution
Texture target network $\theta$
Loss function

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

\[
\mathcal{L}_{\text{texture}}^{(l)}(t, \tilde{x}) = \left( \phi_{\text{texture}}^{(l)}(t) - \phi_{\text{texture}}^{(l)}(\tilde{x}) \right)^2
\]

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

\[
\mathcal{L}_{\text{prior}}(\tilde{x}) = \sum_{i,j} \left( (\hat{x}_{i,j+1} - \tilde{x}_{i,j})^2 + (\hat{x}_{i+1,j} - \tilde{x}_{i,j})^2 \right)^{\beta/2}
\]
Results - Texture Synthesis
Results - Texture Transfer

"knitted"
1920 x 1440 pixels
Results - Texture Transfer

"cracked"
"fibrous"
Results - Texture Transfer

"scaly"
Results - Texture Transfer

"wrinkled"
Thank you!