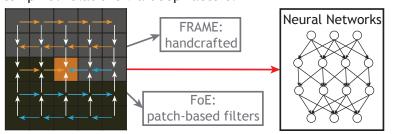
# Deep Markov Random Field for Image Modeling

Zhirong Wu, Dahua Lin, Xiaoou Tang The Chinese University of Hong Kong

http://github.com/zhirongw/deep-mrf

### Motivations

- While MRF provides a generic framework for modeling images, the edge potential is largely limited by simplistic functions.
- We wish to improve the expressive power by modeling the inter-pixel relations via deep factors.



## Deep MRF Formulation

Joint distribution:

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \prod_{u \in V} \zeta(x_u, h_u) \prod_{(u,v) \in E} (\phi(h_u, h_v) \psi(h_u, x_v) \psi(h_v, x_u)) \prod_{u \in V} \lambda(h_u)$$

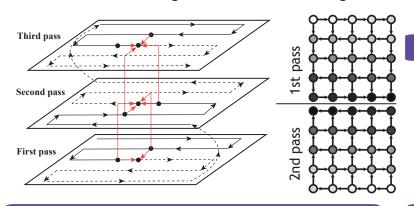
The potential factors:

GMM pixel generation  $\psi(h_u, x_v) = \exp\left(h_u^T \mathbf{R} x_v\right)$ 

GMM pixel generation nearby states 
$$\zeta(x_u,h_u) \triangleq p_{\mathrm{GMM}}(x_u|h_u) \qquad \phi(h_u,h_v) = \exp\left(h_u^T\mathbf{W}h_v\right)$$
 nearby pixel and state states regularizer 
$$\psi(h_u,x_v) = \exp\left(h_u^T\mathbf{R}x_v\right) \qquad \lambda(h_u) = \exp\left(-\mathbf{1}^T\eta(h_u)\right)$$

# **Learning via Coupled Passes**

Decouple a cyclic graph into multiple acylic passes while maintaining full contextual reasoning.



# **Texture Synthesis**

High resolution visually realistic textures!





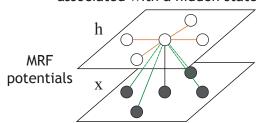






#### Overview

We consider each pixel x is associated with a hidden state h.



- · presents a new powerful MRF model
- theoretical connections between MRF and RNN
- nice results on a variety of low-level applications

### Connections with RNN

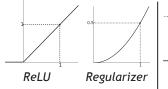
MAP inference of hidden states corresponds to feed-forward computation of an RNN,

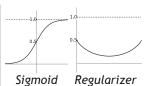
$$\tilde{h}_u = \sigma \left( \sum_{v \in \mathcal{N}_u} \mathbf{W} h_v + \mathbf{R} x_v \right)$$

The activation function of RNN is derived from the regularizer of MRF,

$$\sigma^{-1}(z) = \eta'(z)$$

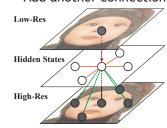
Example activations:





# **Image Super Resolution**

Add another connection from the low-res pixel.



	Performance against CNN			
	PSNR	CNN	SCN	Our
	Set 5	32.30	33.10	33.14
	Set 14	29.00	29.41	29.38
	BSD	28.20	28.50	28.54

## **Image Generation**

