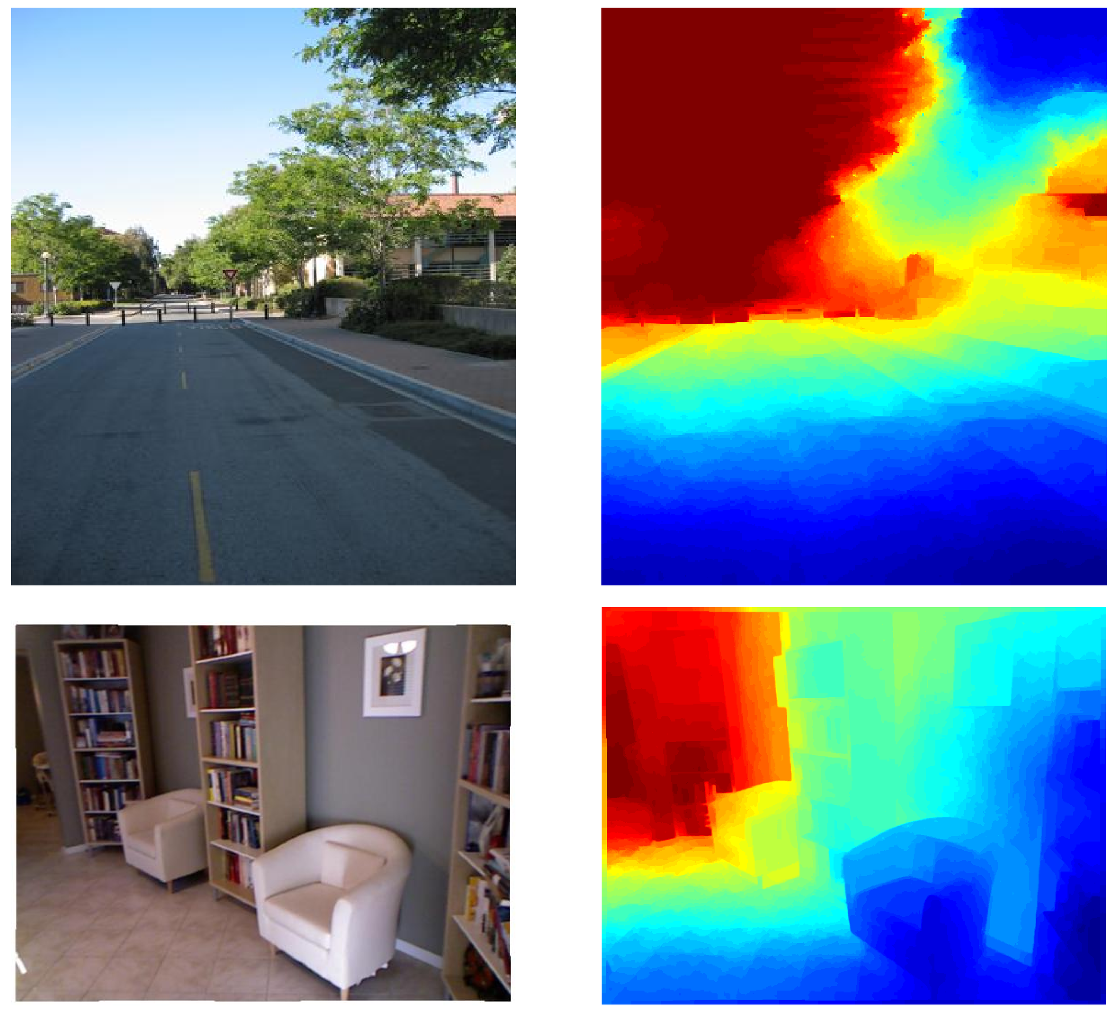
Deep Convolutional Neural Fields for Depth Estimation from a Single Image

Fayao Liu, Chunhua Shen, Guosheng Lin
University of Adelaide, Australia; Australian Centre for Robotic Vision

Introduction

- Depth estimation: estimate depths from single monocular images.
- Challenging: no reliable depth cues, e.g., stereo correspondence, motion information.
- Applications: scene understanding, 3D modelling, robotics, benefit other vision tasks, etc.

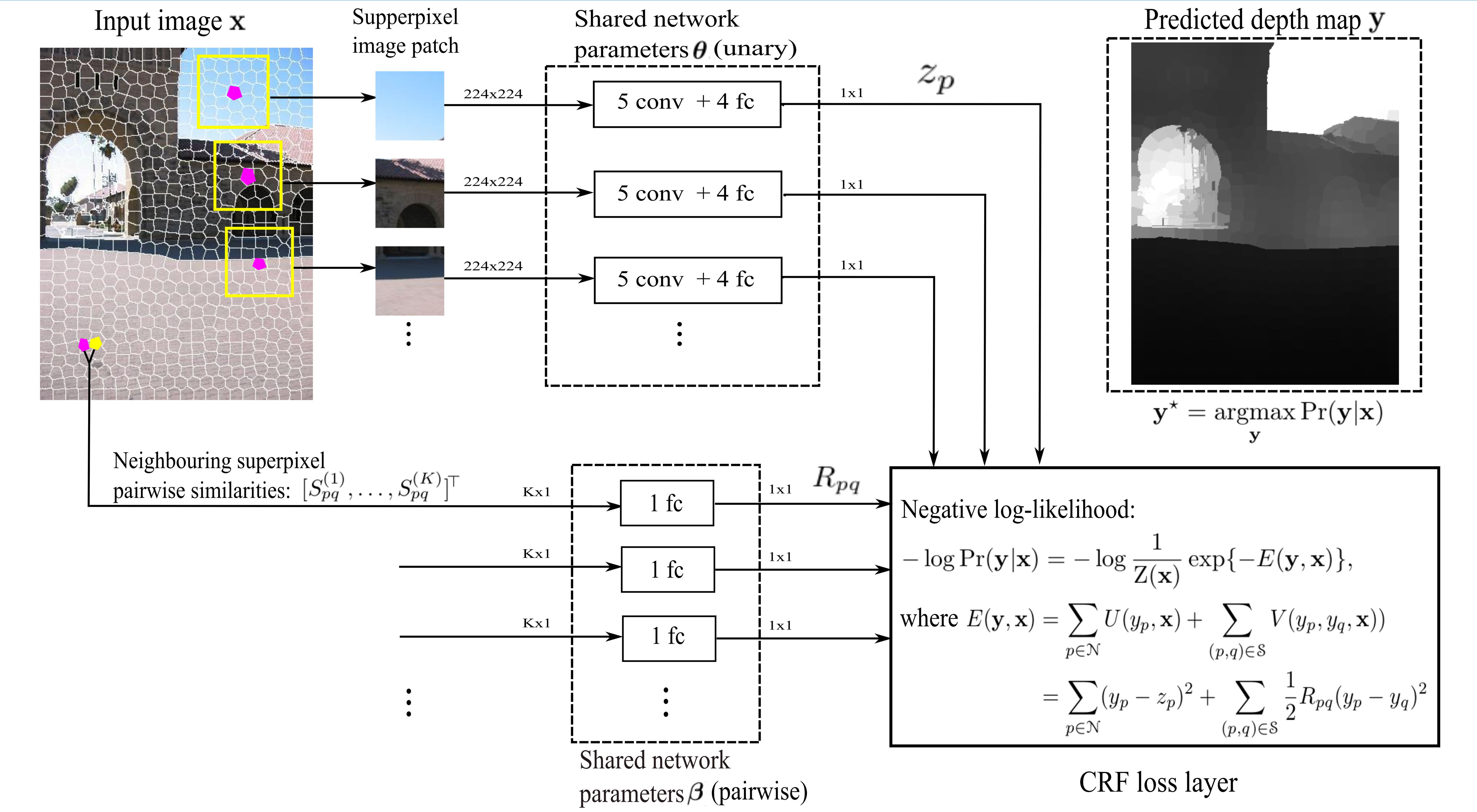


images estimated depth maps

Contributions

- We propose a deep convolutional neural field model for depth estimations by exploring CNN and continuous CRF. Solving the MAP problem for predicting the depth of a new image is highly efficient since closed form solutions exist.
- We jointly learn the unary and pairwise potentials of the CRF in a unified deep CNN framework, which is trained using back propagation.
- We demonstrate that the proposed method outperforms state-of-the-art results of depth estimation on both in- door and outdoor scene datasets.

Method: Deep Convolutional Neural Fields



Continuous CRF

Let $\mathbf{x}$ be an image and $\mathbf{y} = [y_1, \dots, y_n]^\top \in \mathbb{R}^n$ be a vector of continuous depth values of all n superpixels in $\mathbf{x}$. The conditional probability distribution is modelled as:

$$\Pr(\mathbf{y}|\mathbf{x}) = \frac{1}{\mathrm{Z}(\mathbf{x})} \exp(-E(\mathbf{y},\mathbf{x})), \qquad (1)$$

where $\mathrm{Z}(\mathbf{x}) = \int_{\mathbf{y}} \exp\{-E(\mathbf{y},\mathbf{x})\} \mathrm{d}\mathbf{y}$.
The energy function $E(\mathbf{y},\mathbf{x})$ is defined as:

$$E(\mathbf{y},\mathbf{x}) = \sum_{p \in \mathcal{N}} U(y_p, \mathbf{x}) + \sum_{(p,q) \in \mathcal{S}} V(y_p, y_q, \mathbf{x}). \qquad (2)$$

Depth prediction (solve the MAP inference):

$$\mathbf{y}^\star = \operatorname*{argmax}_{\mathbf{y}} \Pr(\mathbf{y}|\mathbf{x}). \qquad (3)$$

Potential Functions

- Unary potential

$$U(y_p, \mathbf{x}; \boldsymbol{\theta}) = (y_p - z_p(\boldsymbol{\theta}))^2. \qquad (4)$$

z_p is the network output of the unary part.

- Pairwise potential

$$V(y_p, y_q, \mathbf{x}; \boldsymbol{\beta}) = \frac{1}{2} R_{pq} (y_p - y_q)^2. \qquad (5)$$

R_{pq} is the output of the pairwise part.

Learning (minimize the negative conditional log-likelihood):

$$\min_{\boldsymbol{\theta}, \boldsymbol{\beta} \geq 0} - \sum_{i=1}^{N} \log \Pr(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}; \boldsymbol{\theta}, \boldsymbol{\beta}) + \frac{\lambda_1}{2} \|\boldsymbol{\theta}\|_2^2 + \frac{\lambda_2}{2} \|\boldsymbol{\beta}\|_2^2, \qquad (6)$$

Experiments

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Make3d	0.349	-	1.214	0.447	0.745	0.897
DepthTransfer	0.35	0.131	1.2	-	-	-
Discrete-continuous CRF	0.335	0.127	1.06	-	-	-
Ladicky *et al.*	-	-	-	0.542	0.829	0.941
Eigen *et al.*	**0.215**	-	0.907	0.611	**0.887**	**0.971**
Ours (pre-train)	0.257	0.101	0.843	0.588	0.868	0.961
Ours (fine-tune)	0.230	**0.095**	**0.824**	**0.614**	0.883	**0.971**
Ours-new (pre-train)	0.234	0.095	0.842	0.604	0.885	0.973
Ours-new (fine-tune)	**0.213**	**0.087**	**0.759**	**0.650**	**0.906**	**0.976**

Table 1: Result comparisons on the NYU v2 dataset.

images ground-truth predictions

Figure 1: Prediction examples on the NYU v2 dataset.

images ground-truth predictions

Figure 2: Prediction examples on the Make3D dataset.