J. Vis. Commun. Image R. 40 (2016) 739–750

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Convolutional neural network based deep conditional random fields for stereo matching ☆

Zhi Wang *, Shiqiang Zhu, Yuehua Li, Zhengzhe Cui

State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, No. 38, Zheda Road, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 28 June 2016
Revised 9 August 2016
Accepted 26 August 2016
Available online 29 August 2016

Keywords:
Stereo matching
Conditional random fields
Convolutional neural network

ABSTRACT

Stereo matching has been studied for many years and is still a challenge problem. The Markov Random Fields (MRF) model and the Conditional Random Fields (CRF) model based methods have achieved good performance recently. Based on these pioneer works, a deep conditional random fields based stereo matching algorithm is proposed in this paper, which draws a connection between the Convolutional Neural Network (CNN) and CRF. The object knowledge is used as a soft constraint, which can effectively improve the depth estimation accuracy. Moreover, we proposed a CNN potential function that learns the potentials of CRF in a CNN framework. The inference of the CRF model is formulated as a Recurrent Neural Network (RNN). A variety of experiments have been conducted on KITTI and Middlebury benchmark. The results show that the proposed algorithm can produce state-of-the-art results and outperform other MRF-based or CRF-based methods.

1. Introduction

Stereo matching, as a long-standing problem in the research of computer vision [1], has been used in a variety of fields, such as 3d reconstruction, navigation for autonomous driving system, panoramic stereo imaging [2] and DoF rendering [3]. There are generally two kinds of techniques for stereo matching: local methods (also known as cost filtering methods) and global methods (also known as energy minimization methods). The cost filtering methods are widely used in the early times because of its simple implementation. Many local stereo matching algorithms have been presented, such as bilateral filter method (BF) [4], guided filter method (GF) [5], non-local cost aggregation method [6]. The core idea of this kind of methods is to determine the disparity of the pixel based on the neighbouring pixels around it [7]. The local methods usually make the assumption that all pixels in the local support window have similar disparities. The size of the support window largely affects the matching accuracy. The global methods have drawn much attention recently due to the development of the energy minimization methods in image segmentation [8]. Early implementations of global methods include graph cut [9], dynamic programming [10] and belief propagation [11]. The disparity maps obtained by the global methods are generally more dense and accurate than local methods.

1.1. Related works

The global methods usually determine the disparity results through minimizing the MRF-based energy function. The slanted-plane MRF model for stereo matching was first introduced by Birchfield and Tomasi [12]. However, the smoothness in this study was defined very locally. Thus, many improvements have been made on it to achieve better performance [13,14]. Yang et al. [13] proposed a real-time color segmentation method and used BP-based plane-fitting algorithm for stereo matching. Bleyer et al. [14] presented a pixel-wise MRF model, in which each pixel is assigned to a B-spine surface. The Conditional Random Field (CRF) was first proposed for sequence processing by Lafferty et al. [15]. It has been widely used in image labelling and segmentation problems. Pal et al. [16] extended the CRF model to stereo matching which adopted the gradient-based learning method.

In the earlier studies, the stereo matching problem was usually formulated as the energy minimization problem without an explicit model, like the partial differential equation (PDE) [17] and the maximum a posteriori (MAP) solution [18]. Thus, there was no optimization for the stereo models. When the traditional stereo algorithms are cast in a CRF model, the learning can be implemented through conditional likelihood estimation. However, when the images are from the real environments, the models become

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail address: 11325067@zju.edu.cn (Z. Wang).

http://dx.doi.org/10.1016/j.jvcir.2016.08.022
1047-3203/

more challenging. This makes the learning more challenging. The main challenge for learning a CRF lies at the computation of the model expectations in the energy function. To solve this problem, the graph-cut-based method for energy minimization was proposed by Szeliski et al. [19], which was fast and accurate.

Video coding [20,21] is also actively researched in computer vision. High-efficiency video coding (HEVC) [22,23] is the next generation standard of video coding. In this paper, we focus on the research of stereo matching. There has been increasing in interest in the machine learning for stereo matching in recent years. Due to the limitations of training data sets, only a few approaches were presented. The most typical one is the study of Zhang and Seitz [24], in which the parameters of the MRF model were learnt from the previous estimated disparities. Trinh and McAllester [25] introduced a CRF-like model for stereo matching that the CRF model with hidden variables were constructed for depth estimation. Recently, Convolutional Neural Network has been developing fast in stereo matching as the advances in models learning and training data sets. Žbontar and LeCun [26] trained a CNN to compute the matching cost and achieved good performance. Chen et al. [27] proposed a Deep Embedding model to train the CNN and estimate the disparities.

1.2. Motivations

In this paper, we demonstrate that the local methods cannot solve two kinds of ambiguities: two pixels with similar color but at different disparities (also called as depth ambiguity), two pixels with different colors but at the same disparity (also called as color ambiguity), as shown in Figs. 1 and 2.

In the case of filter-based local methods, or the first ambiguity, the weights of two pixels computed by the filters are equal because of the same spatial distance and color similarity. It is obvious that these two weights should not be the same. Thus, the filter-based local methods fail to represent the difference disparities between p and q. As to the second ambiguity, many existing segment-based local methods [28,29] are ineffective as they usually adopt low-level segmentation methods, which divide the same disparity region into two areas with different constraints. These methods can hardly distinguish the depth discontinuity from the color discontinuity. To solve this problem, Bleyer et al. [30] proposed an object stereo method that implemented the stereo matching and segmentation at the same time. Similarly, Xu et al. [18] presented a PM-PM method for joint segmentation and stereo matching. Based on these observations, the object knowledge is adopted as a soft constraint to sort the depth and color ambiguities.

The main concerns for the CRF model based stereo are the learning and inference. Although the graph-cut-based method for the conditional likelihood computation is fast and effective, it has certain limitations. Thus, the graph-cut inference may fail when learning some models. The belief propagation methods [31] are widely used in many applications in computer vision and are known for its easy implementation. Based on this work, many improvements have been made. Ihler and Mcallester [32] presented the particle belief propagation. Hazan and Shashua [33] proposed the norm-product belief propagation. Due to the development of deep learning technology, CNN has been successful in high-level tasks such as object detection [34] and semantic segmentation [35]. However, the CNN has convolutional filters and lacks smoothness constraints, which would produce coarse outputs and non-sharp boundaries. The CRF inference can refine the coarse outputs and produce sharp boundaries, which is able to solve the problems in utilizing CNN. Recently, a variety of methods have demonstrated how to joint train the graphical models and classifiers [36]. Thus,

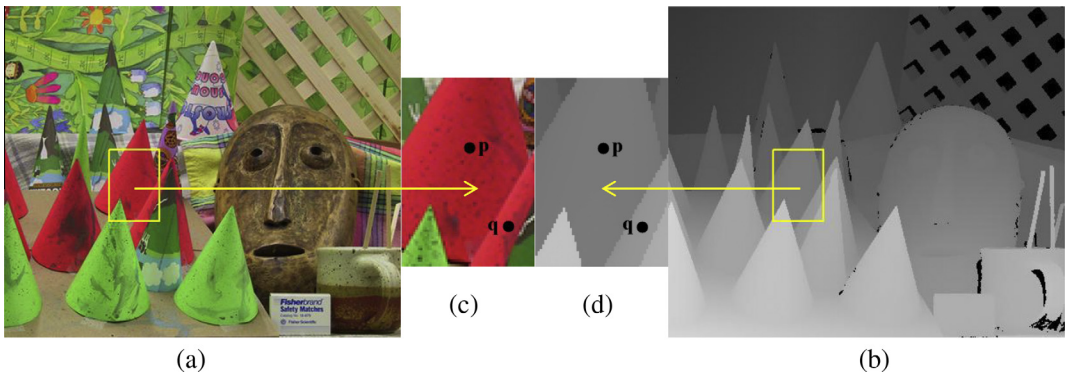

Fig. 1. Depth ambiguity. (a) Reference image. (b) Depth image. (c) Two nearby pixels with similar colors but at different disparities. (d) Zoomed-in view of (b).

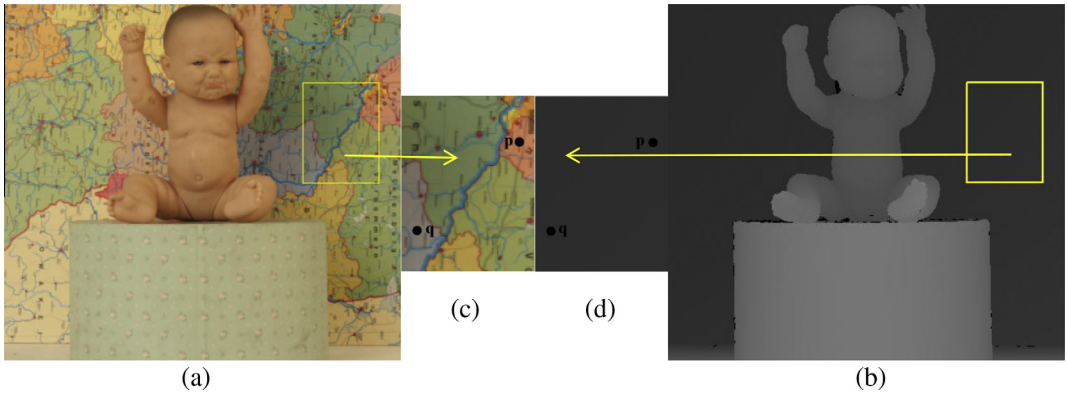

Fig. 2. Color ambiguity. (a) Reference image. (b) Depth image. (c) Two nearby pixels with different colors but at the same disparity. (d) Zoomed-in view of (b).

in this study, the CNN is combined with the CRF model for inference.

After surveying and analyzing the existing stereo matching algorithms, we proposed an object-level deep CRFs (DCRF) based method for stereo matching. In this paper, two connections are drawn between the object segmentation and stereo matching, the CNN and CRF, respectively. First, we initialize all superpixels using StereoSLIC algorithm [37] based on the assumption that each image is composed of small homogeneous regions. The ALE [38] is used for object segmentation. Then, the object model is employed as a soft constraint in the CRF formulation, which can produce more accurate depth estimations. The CRF inference is formulated as a Recurrent Neural Network (RNN) [39]. Finally, the multi-step refinements are taken to obtain accurate disparity maps. The details are shown in Fig. 3. Experiments have been conducted on the challenging KITTI and Middlebury benchmark, and the experimental results demonstrate the effectiveness and advantage of the proposed method.

1.3. Organization of the paper

The organization of the paper is shown as follows. Section 2 introduces the deep CRF model. The CRF combined with RNN for inference is proposed in Section 3. Section 4 presents the systematic multi-step disparity refinement algorithm. The experimental results and its analysis are given in Section 5. Section 6 gives the conclusion.

2. The deep CRF model for stereo matching

In this section, we first briefly introduce the general CRF architecture for stereo. Then, we present the proposed deep CRF model.

Consider a random field $I = \{I_1, I_2, \ldots, I_N\}$, where I_j denotes the color vector of pixel j. Consider also a random field $X = \{X_1, X_2, \ldots, X_N\}$, where X_i takes the values from label space $\ell = \{d_1, d_2, \ldots, d_m\}$, which corresponds to the set of disparities. Then, a conditional random field can be expressed as:

$$P(X|I) = \frac{1}{Z(I)} \exp\left(-E(X,I)\right), \tag{1}$$

with

$$Z(I) = \sum_{X} \exp(-E(X,I)), \tag{2}$$

where $E(X,I)$ represents the energy function taking the form as:

$$E(X,I) = \sum_{i} \psi_u(X_i, I) + \sum_{ij} \psi_p(X_i, X_j, I), \tag{3}$$

The unary potentials $\psi_u(X_i, I)$ are computed through the color dissimilarity between the corresponding pixels given a disparity X_i. The pairwise potentials $\psi_p(X_i, X_j, I)$ encourage local smoothness by encouraging the neighbouring pixels to take similar disparity. Then, the stereo matching problem can be formulated as the minimization of the energy function.

The CRF specified in (3) is the general form. Here we introduce the proposed CRF model for disparity estimation. Let R denote the set of all pixels. Let C denote the label space $C = \{c_1, c_2, \ldots, c_l\}$. Each label c_k indicates the object class. Inspired from [40], each image point is expressed as $p = [x, y, 1]^T$. And a random variable $n = [a, b, c]^T$ is defined to describe the random plane in 3D. The plane π is represented as $\pi = [n^T, f]$ with $f = -ax - by - cd$, where d is disparity of p. Thus, for the point $q = [x_q, y_q, 1]^T$ lying on the plane π, the disparity is given by:

$$d_q = -\frac{a}{c}(x_q - x) - \frac{b}{c}(y_q - y) + d, \tag{4}$$

The energy function in this paper is defined as:

$$E(X,n) = \sum_{i \in R} \psi_i(X_i, n_i) + \sum_{ij} \psi_{ij}(X_i, X_j) + \sum_{c \in C, i \in R} \psi_c(n_i) + \sum_{i \in R} \psi_{CNN}(X_i, \xi), \tag{5}$$

2.1. Data term

The data term (cost) represents the similarity between the corresponding pixels. A variety of cost computation methods have been studied, including the sum of absolute difference (AD), the Rank transform (Rank) and normalized cross correlation (NCC). In this study, we use the combination of the raw input image and the enhanced image processed by the weighted guided filter (WGIF) [41]. There are mainly two kinds of cost measurements: the pixel-to-pixel matching dissimilarity and non-parametric transforms such as census and rank. Using the combined strategy to compute the matching cost is superior to the single one [42]. In this paper, a 6D feature vector function is defined to include both the information of the raw and enhanced image. The function is calculated as:

$$I_f = \left(I_R^{raw}, I_G^{raw}, I_B^{raw}, I_R^E, I_G^E, I_B^E\right), \tag{6}$$

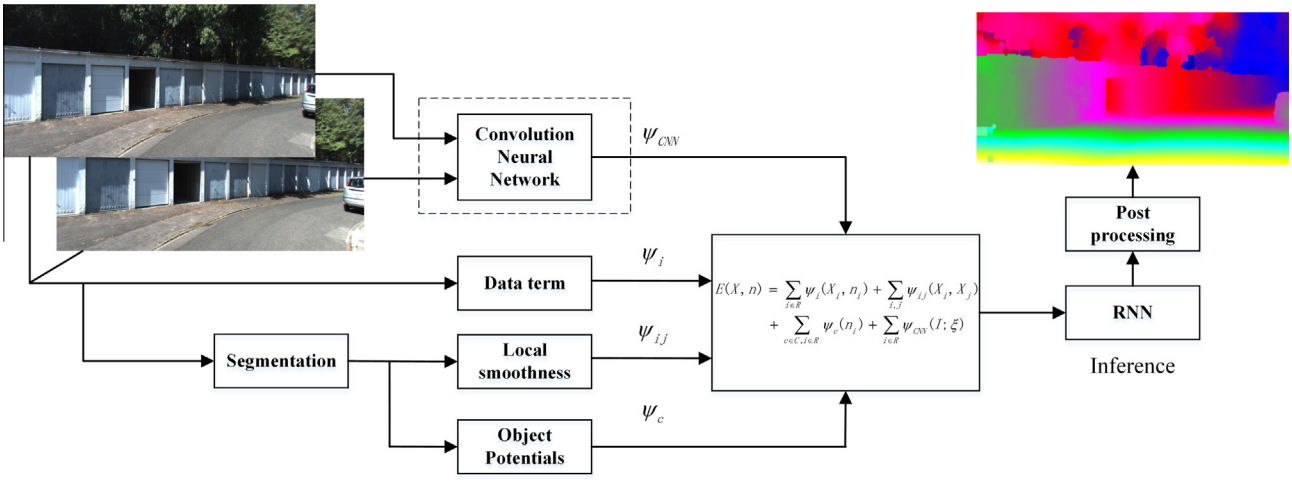

Fig. 3. The architecture of our DCRF model for stereo matching.

$$grad_{dir} = \left(\nabla_{dir} I_R^{raw}, \nabla_{dir} I_G^{raw}, \nabla_{dir} I_B^{raw}, \nabla_{dir} I_R^{E}, \nabla_{dir} I_G^{E}, \nabla_{dir} I_B^{E}\right), \tag{7}$$

where I_f is the color vector and $grad_{dir}$ is the gradient vector. *raw* and *E* represent the raw and enhanced image, respectively. *dir* indicates the *x* or *y* axis. The 6D vector function contains both global boundaries information and local details, thus it is more robust and outperforms the traditional cost computation methods.

The pixel similarity in our study is defined as:

$$\rho(X_i, n_i) = \alpha\psi_i^I(X_i, n_i) + \beta\psi_i^G(X_i, n_i) + \gamma\psi_i^{cen}(X_i, n_i), \tag{8}$$

where

$$\psi_i^I(X_i, n_i) = \min(\|I_f(i) - I_f(i - (ax_i + by_i + c))\|, \tau_{col}), \tag{9}$$

$$\psi_i^G(X_i, n_i) = \min(\|grad_x(i) - grad_x(i - (ax_i + by_i + c))\| + \|grad_y(i) - grad_y(i - (ax_i + by_i + c))\|, \tau_{grad}), \tag{10}$$

$$\psi_i^{cen}(X_i, n_i) = \min(\text{Hamming}(cen(i), cen(i - (ax_i + by_i + c)), \tau_{cen}), \tag{11}$$

$\psi_i^I(X_i, n_i)$ and $\psi_i^G(X_i, n_i)$ represent the color similarity and the gradient similarity, respectively. $\psi_i^{cen}(X_i, n_i)$ denotes the measurement using census transform. α, β, γ are the weight parameters. $\tau_{col}, \tau_{grad}, \tau_{cen}$ are the threshold of each similarity function. $cen(i)$ is the census transform which is specified by:

$$cen(i) = \otimes_{j \in N(i)} bin(i, i - (ax_i + by_i + c)), \tag{12}$$

where $bin(i, i - (ax_i + by_i + c))$ is a binary function and $\otimes$ represents concatenation. $N(i)$ is a square window centered at *i*.

Similar to the work in [40], we calculate the data term using the aggregated matching cost, which is defined as:

$$\psi_i(X_i, n_i) = \sum_{j \in N(i)} \omega(i,j)\rho(X_j, n_j), \tag{13}$$

where $\omega(i,j)$ is a weight function that computes the weight between *i* and *j*. The weight function is expressed as:

$$\omega(i,j) = \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right), \tag{14}$$

where $\theta_\alpha, \theta_\beta$ are intrinsic parameters to control the degree of similarity.

2.2. Local smoothness

The smoothness term enforces the consistency of object label and disparity function between the adjacent pixels. As the consistency of disparity function is related to that of object label, the object boundary seems more possibly to appear in the disparity inconsistent regions. Thus, we decompose the smoothness term as:

$$\psi_{ij}(X_i, X_j) = \theta_1\psi_{ij}^o(X_i, X_j) + \theta_2\psi_{ij}^m(X_i, X_j), \tag{15}$$

where θ_1 and θ_2 are two weights that control the contribution of each term.

$$\psi_{ij}^o(X_i, X_j) = \omega(i,j)\varphi(X_i, X_j), \tag{16}$$

where $\varphi(X_i, X_j)$ is a penalty function, which is defined as:

$$\varphi(X_i, X_j) = \begin{cases} 0, & if(X_i = X_j) \\ 1, & otherwise \end{cases}, \tag{17}$$

The pairwise term $\psi_{ij}^m(X_i, X_j)$ represents the deviation between two disparity planes, which is given by:

$$\psi_{ij}^m(X_i, X_j) = \omega(i,j)(|v_i \cdot (P_i - P_j)| + |v_j \cdot (P_j - P_i)|), \tag{18}$$

where $v_i = orth([a, b, -1]^T)$, $orth = v/\|v\|$, and $P_i = [x_i, y_i, c]^T$.

2.3. Object potentials

The conditional random filed formulation described in (3) was extended in [43] by incorporating higher-order potentials. It made the assumption that pixels located within the same segment label are more possible to take the same disparity value. The higher-order term was of the form according to [43]:

$$\psi_c^h(X_k) = \min_{k \in \ell}\left(\eta_c^{\max}, \eta_c^k + \sum_{i \in c} \omega_i \gamma_c^k \Delta(X_i \neq k)\right), \tag{19}$$

where η_c^k is the cost representing that all pixels with the same segment label have the same disparity. If the pixel does not take the disparity, it is penalized with a cost $\omega_i\gamma_c^k$, whereω_i is the weight of X_i. $\eta_c^{\max}$ is the maximum value of η_c^k, which enforces the integration of multiple segmentations.Furthermore, we define an object potential between each object class and all pixels it comprises. It is defined as follow:

$$\psi_{ci}^p(n_i) = \lambda_{ci}[l_c = 1] \cdot (1 - \delta(n_i, n_{c,\pi_i})), \tag{20}$$

where l_c denotes the absence (l_c = 0) or presence (l_c = 1), and $[\cdot]$ is the Iverson bracket. λ_{ci} is a penalty value. The function $\delta(n_i, n_{c,\pi_i})$ is expressed as:

$$\delta(n_i, n_{c,\pi_i}) = \begin{cases} 1, & if\ n_i = n_{c,\pi_i} \\ 0, & otherwise \end{cases}, \tag{21}$$

Thus, the object potentials term in this paper is given by:

$$\psi_c(n_i) = \psi_c^h(X_k) + \psi_{ci}^p(n_i), \tag{22}$$

2.4. CNN potential function

The CNN potential function is formulated by a CNN. This function is defined as:

$$\psi_{CNN}(X_{i,}, \xi) = -z_i(X_i, \xi)^2, \tag{23}$$

where ξ is the parameter of CNN, and z_i is the output of CNN. The CNN architecture is depicted in Fig. 4. Our CNN model is mainly based on the central-surround two-stream network [44] with modifications. All convolutional layers are equipped with rectified linear units (ReLU) [45]. The implementation details are shown as follow:

(1) The inputs are two $13 \times 13 \times 1$ image patches, and the red-colored dash box represents a $\times 2$ down-sampling. The first layer and second convolutional layer consist of 32 kernels of size $5 \times 5 \times 1$. Layer *L*3 is applied with filters of size $5 \times 5 \times 32$ and layer *L*4 is applied with filters of size $1 \times 1 \times 200$. The output of these four layers are computed only once per pixel *i* regardless of the disparity.

(2) The output of *L*4 is concatenated into a 400 dimensional vector and passed through the fully connected layer *L*5 and *L*6. The score of each network is obtained by the softmax function. The two matching scores are merged by a $1 \times 1 \times 2$ convolutional layer to obtain the final output.

3. Inference in the CRF model

Inspired by the work in [46] that jointly trains the CNN and CRF, we explore the use of CRF combined with CNN for depth estimation. The final disparity is obtained by minimizing the CRF energy function. In most studies, the mean field approximation was adopted for inference, which used the iterative message passing method. In this paper, the mean field approximation is treated as a CNN and trained end-to-end with back-propagation. In this section, we first introduce the standard mean field approximation, and then present the proposed inference method.

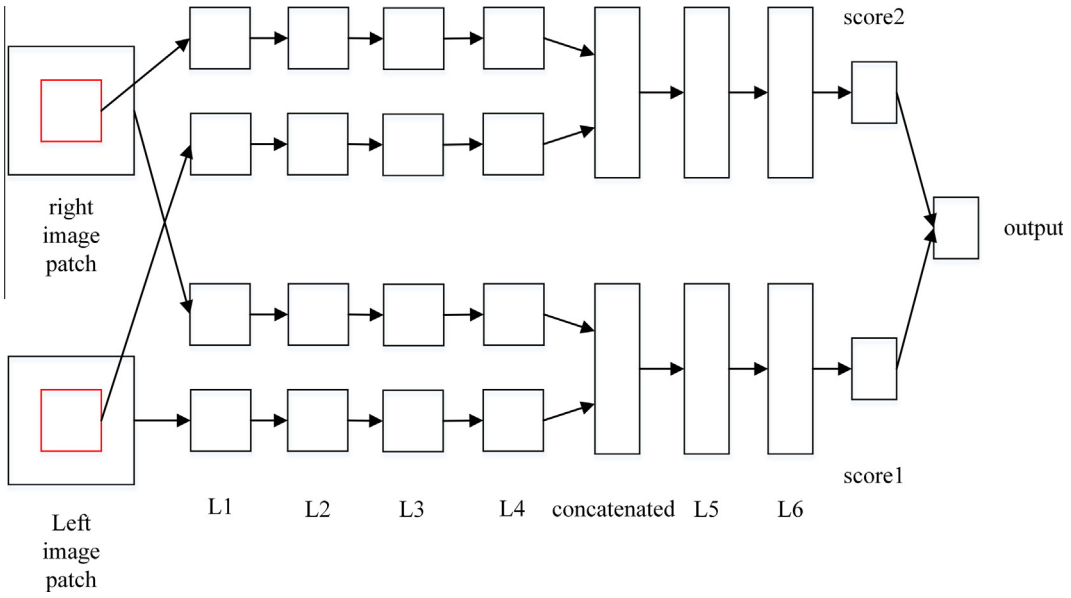

Fig. 4. The detailed network architecture of the CNN potential function.

3.1. Mean field approximation

The mean field approximation is usually implemented by computing the distribution $Q(X)$ that minimizes the KL divergence, which is expressed as follow:

$$\begin{aligned} KL(Q(X)\|P(X|I)) &= \sum_X Q(X)\log\frac{Q(X)}{P(X|I)} \\ &= \sum_X Q(X)\log\frac{Q(X)Z(I)}{\exp(-E(X,I))} \\ &= \sum_X Q(X)E(X,I) + \sum_X Q(X)\log Q(X) + \log Z(I), \end{aligned} \tag{24}$$

We define a free energy term to be:

$$\begin{aligned} \Gamma(Q(X)) &= \sum_X Q(X)E(X,I) + \sum_X Q(X)\log Q(X) \\ &= F(E(X,I)|Q(X)) - H(Q(X)), \end{aligned} \tag{25}$$

where $\mathrm{F}(E(X,I)|Q(X))$ is the expected energy under the distribution $Q(X)$ and $H(Q(X))$ is the entropy of $Q(X)$. Then, the divergence becomes:

$$KL(Q(X)\|P(X|I)) = \Gamma(Q(X)) + \log Z(I), \tag{26}$$

Since $\log Z(I)$ keeps constant for the given I, minimizing the KL divergence is equal to minimize the free energy $\Gamma(Q(X))$. As the distribution $Q(X)$ can be expressed as the product of margins:$Q(X) = \prod_i Q(X_i)$, then the minimization of the KL divergence can be expressed as:

$$\begin{aligned} \Gamma(Q(X)) &= \sum_X \prod_i Q(X_i)E(X,I) + \sum_X \prod_i Q(X_i)\log Q(X) \\ &= \sum_X Q(X_j)\mathrm{F}(E(X,I)\Big|\prod_{i,i\neq j} Q(X_i)) - H(Q(X_j)) - \sum_{i,i\neq j} H(Q(X_i)), \end{aligned} \tag{27}$$

We use the Lagrange multipliers to constrain the $Q(X)$ and $Q(X_i)$ to be valid, which yields iterative update equation that computes the updated approximating distribution:

$$Q^*(X_j = d_j) = \frac{1}{Z_j}\exp\left(\mathrm{F}\left(E(X,I)\Big|\prod_{i,i\neq j} Q(X_i)\right)\right), \tag{28}$$

where Z_j is a constant so that the sums of $Q^*(X_j)$ is one. Updating the $Q(X_j)$ will decrease the free energy. Thus the KL divergence is minimized in this manner. But when the state space is large, the calculation will become extremely complex. To solve this problem, the sparse variational message passing [16] was proposed. The main idea is to eliminate the values of the disparity from consideration. Based on this observation, the KL divergence can be defined as the function of the original update $Q^*(X_j)$,the sparse update $Q'(X_j)$ and other distributions $Q(X_i)$.

$$KL(Q(X)\|P(X|I)) = KL(Q'(X_j)\|Q^*(X_j)) + \log Z_j + \log Z(I) - \sum_{i,i\neq j} H(Q(X_i)) \tag{29}$$

The minimization of $KL(Q(X)\|P(X|I))$ is equal to that of $KL(Q'(X_j)\|Q^*(X_j))$ as the last 3 terms keep constant. The minimization is implemented when $Q'(X_j) = Q^*(X_j)$.

3.2. End-to-end network for inference

The message passing step of the mean field approximation usually relies on the Gaussian filters. But the coefficients of the Gaussian filters are derived from the image features such as the color similarity and spatial distance in the RGB space. Since the CRF is fully connected, the brute-force implementation of these filters is infeasible. Recently, several high-dimensional filtering methods have been proposed to make the computation faster [47]. The computational complexity of these methods is $O(N)$, where N is the number of all image pixels. Thus, the back-propagation can be also implemented in $O(N)$ time. Based on this, in this paper, we formulate the message passing step as a Recurrent Neural Network (RNN) [39]. Using the same notation as [48], for a given input image patch I_k, the output of a network with M stages can be written as:

$$f(I_k,(W,b)) = W_M H_{M-1}, s \tag{30}$$

where H_m is the output of the mth hidden layer ($m = 1, \ldots, M$), which is computed as:

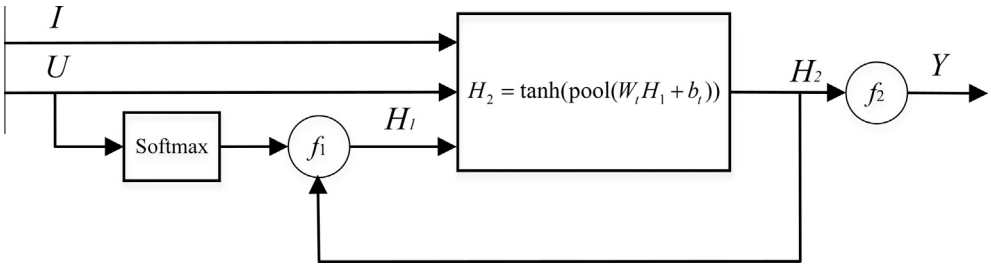

Fig. 5. The RNN architecture for inference. The functions f_1 and f_2 are described in (32) and (34).

$$H_m = \tanh(\text{pool}(W_m H_{m-1} + b_m)), \tag{31}$$

where b_m denotes the bias vector of the mth layer and W_m represents the Toeplitz matrix of connection between the mth and $(m-1)$th layer. The mean field iterations mentioned above can be implemented by a stack of layers, in which the update Q is estimated from the previous layer. The first network takes the input as predictions. This is equivalent to formulate the mean field approximation as a RNN, which is shown in Fig. 5. The iterations can be expressed as:

$$H_1 = \begin{cases} \text{softmax}(U), & t = 0 \\ H_2(t-1), & 0 < t \leqslant T \end{cases}, \tag{32}$$

$$H_2 = \tanh(\text{pool}(W_t H_1 + b_t)), \quad 0 \leqslant t \leqslant T, \tag{33}$$

$$Y(t) = \begin{cases} 0, & 0 \leqslant t \leqslant T \\ H_2(t), & t = T \end{cases}, \tag{34}$$

where t = {0, ..., T}. T is the number of iterations and U denotes the potential values. The learning for the RNN is the same as the standard CNN, in which the stochastic gradient descent (SGD) algorithm is used. The network is trained by applying the *softmax* function:

$$p(\eta|I_k;(W,b)) = \frac{\exp(f_\eta(\eta|I_k;(W,b)))}{\sum_{d\in\{d_1,\ldots,d_m\}}\exp(f_d(d|I_k;(W,b)))}, \tag{35}$$

where $\eta \in \{d_1, \ldots, d_m\}$. The parameters (W, b) can be learnt by minimizing the negative log-likelihood of the training data [16].

$$L_f(W,b) = -\sum_{I_k} \ln p(\eta|I_k;(W,b)), \tag{36}$$

The minimization could be achieved through the SGD method with a fixed learning rate λ.

$$\begin{aligned} W - \lambda\frac{\partial L_f}{\partial W} &\to W \\ b - \lambda\frac{\partial L_f}{\partial b} &\to b \end{aligned} \tag{37}$$

Then, the disparity label for each pixel can be predicted as:

$$\widehat{\eta} = \text{argmax}_{d\in\{d_1,\ldots,d_m\}} p(\eta|I_k;(W,b)), \tag{38}$$

4. Disparity refinement

As the raw disparity maps contain many outliers, thus, the disparity refinement step is an essential part for stereo matching to produce good results. All state-of-the-art stereo matching algorithms adopt post processing step. There are mainly two ways to implement the refinement: 1. Use the disparity computation method to obtain the raw disparities and then post process it; 2. Merge the post processing step with the core part of the method, like the adaptingBP [49], doubleBP [50]. We take the former method. However, it is hard to remove all the invalid disparities using only one post-processing method. So the multi-step post processing method is adopted in this paper to fully remove the outliers. In our case, the post processing includes the left-to-right consistency (LRC) check, four-direction propagation, the gradient domain guided image filtering.

4.1. LRC check

In this study, the pixels are categorized as: stable pixels, outliers. In the beginning, the LRC check is adopted to determine the invalid pixels in the raw disparity maps. Let d_L, d_R denote the left and right disparity value, respectively. LRC check makes the assumption that the disparity value of the pixel $p(x,y)$ in the left image is equal to that of the corresponding pixel $q(x - \max(d_L, 0), y)$:

$$d_L(x,y) = d_R(x - \max(d_L, 0), y), \tag{39}$$

If these two values are not equal, the pixel is labelled as an outlier.

4.2. Four-direction propagation

After the outliers are detected, the four-direction propagation strategy is used to correct the outliers. The main idea is to search for the nearest valid pixel along horizontal and vertical directions. In many previous studies, the scan-line propagation is adopted, but it cannot work well in some cases. For example, for the leftmost outliers, the scan-line propagation fails due to the lack of corresponding pixels in the right image. Thus, we extend it to four directions, which can be expressed as:

$$d_p = \begin{cases} d_{lr}, & \text{if only } d_{lr} \text{ existing} \\ d_{ud}, & \textit{else} \text{ if only } d_{ud} \text{ existing} \\ null, & \textit{else} \text{ if only } d_{ud} \text{ and } d_{lr} \text{ not existing} \\ \min(dlr, dud), & otherwise \end{cases}, \tag{40}$$

where $d_{lr} = \min(d_l, d_r), d_{ud} = \min(d_u, d_d)$, and $\{d_l, d_r, d_u, d_d\}$ represent the disparity value of the nearest valid pixel in the four directions, respectively.

4.3. Gradient domain guided image filter

Unlike most studies that adopt the adaptive support weight function [4] or weighted median filter for post processing, in this paper, we use the gradient domain guided image filter (GIF) [51] shown as follow:

$$\omega(i,j) = \begin{cases} \frac{1}{|w|^2}\sum_{k:(i,j)\in N(k)}\left(1 + \frac{(I_f(i)-\mu_k)(I_f(j)-\mu_k)}{\sigma_k^2 + \frac{\lambda}{\Gamma_G(k)}}\right), & \text{if } j \text{ is stable} \\ 0, & \text{else} \end{cases}, \tag{41}$$

where $|w|$ is the pixel number in the window $N(k)$. μ_k is the mean value of I in $N(k)$ and σ_k are the variance of I in the window $N(k)$.

λ is a regularization parameter and $\hat{\Gamma}_G(k)$ is a weight. The gradient domain GIF has better edge-preserving property and lower computational complexity. Thus, it is more accurate in stereo matching than the adaptive support weight based method [52]. Then the updated disparity can be expressed as:

$$d_g = \text{argmax}_{d\in\{d_1,\ldots,d_m\}}\left(\sum_{j\in N(i)} w(i,j)\times f(j,d)\right), \tag{42}$$

where

$$f(i,d) = \begin{cases} 1, & \text{if } d(j) = d \\ 0, & \text{else} \end{cases}, \tag{43}$$

5. Experimental results

Comprehensive experiments have been conducted to obtain a thorough quantitative analysis of the proposed algorithm. The state-of-the-art methods are also used for comparison. We mainly focus on KITTI dataset for training and testing because it has large number of images and is more challenging. Moreover, the Middlebury dataset is also used for evaluation. The Neural Network architecture of our system is trained by using the standard Caffe [53] deep learning resource.

5.1. Energy term evaluation

The first set of experiments is conducted on Middlebury and KITTI dataset to assess the contribution of each term in the energy function. At first, the contribution assessment of each term is implemented. There are four terms in our energy function: data term ("data"), local smoothness ("smo"), object potentials ("obj") and CNN potential function ("CNN"). The data term makes the assumption that the corresponding pixels should have similar appearance. The local smoothness term encourages the neighbouring pixels to have the same disparity. The object potentials term assumes that pixels in the same object class are more likely to take the same disparity value. The CNN potential function measures the similarity of the corresponding pixels. To demonstrate the advantage of the proposed energy function, a quantitative comparison is made on the four stereo pairs (i.e. Venus, Tsukuba, Teddy, Cones) as shown in Fig. 6. These results are raw matching results without any optimization and refinements. From Fig. 6, it can be found that the proposed energy function outperforms the combinations of

	A	B	C	D	E
Venus	3.23	2.08	1.53	1.31	0.77
Tsukuba	4.08	2.92	2.12	1.92	1.21
Teddy	11.86	6.89	5.83	5.45	4.68
Cones	7.22	4.37	3.39	3.07	2.85
Average	6.6	4.01	3.22	3.19	2.34

Fig. 6. Quantitative comparison of different combinations of each term of the energy function. *nonocc* means non-occluded area. A: data, B: data + smo, C: data + smo + obj, D: data + smo + CNN, E: data + smo + obj + CNN.

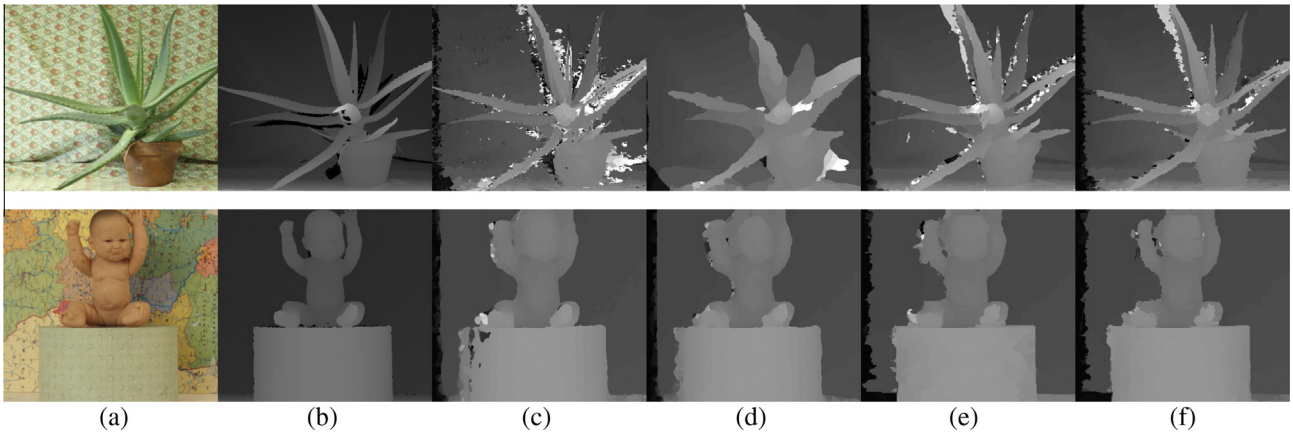

Fig. 7. Disparity results of different energy functions. (a) Input image, (b) ground truth, (c) data, (d) data + smo, (e) data + smo + obj, (f) data + smo + obj + CNN.

Table 1
Importance of each term in the energy function.

Energy term	Out-Noc (%)	Out-All (%)
Data	5.33	6.28
Data + smo	4.82	5.99
Data + smo + obj	4.27	5.13
Data + smo + CNN	4.08	4.87
Data + smo + obj + CNN	3.65	4.61

each term. The data term alone has the worst performance with an average error of 6.6%. The local smoothness term is able to reduce the error percentage by 2.59%. Adding the object potential term further reduces the error due to the fact that the object constraint can solve the ambiguity problems mentioned in Section 1.2. Finally, combining the four terms in the energy function obtains the best performance with an average error of 2.34%. The visual comparisons are also made on the Aloe and Baby1 stereo pairs of

Table 2
Experimental results on the KITTI benchmark with error threshold 3.

Method	Out-Noc (%)	Out-All (%)	Avg-Noc (px)	Avg-All (px)	Runtime	Environment
DCRF	3.29	4.32	0.9	1.3	38 s	Nvidia GTX 970 (CUDA, Caffe)
CoR [57]	3.30	4.10	0.8	0.9	6 s	6 cores @ 3.3 GHz (Matlab C/C++)
SPS-St [56]	3.39	4.41	0.9	1.0	2 s	1 core @3.5 GHz (C/C++)
DDS-SS [58]	3.84	4.59	0.9	1.0	1 min	1 core @2.5 GHz (C/C++)
PCBP [54]	4.04	5.37	0.9	1.1	5 min	4 core @2.5 GHz (C/C++)
CSPMS [59]	4.13	5.92	1.2	1.6	6 s	4 core @2.5 GHz (C/C++)
MBM [60]	4.35	5.43	1.0	1.10	0.2 s	1 core @3.0 GHz (C/C++)
rSGM [55]	5.03	6.60	1.1	1.5	0.2 s	4 core @2.6 GHz (C/C++)
ARW [61]	5.20	6.87	1.2	1.5	4.6 s	1 core @3.5 GHz (C/C++)
GF [5]	11.65	13.76	4.5	5.6	120 s	1 core @3.0 GHz (C/C++)

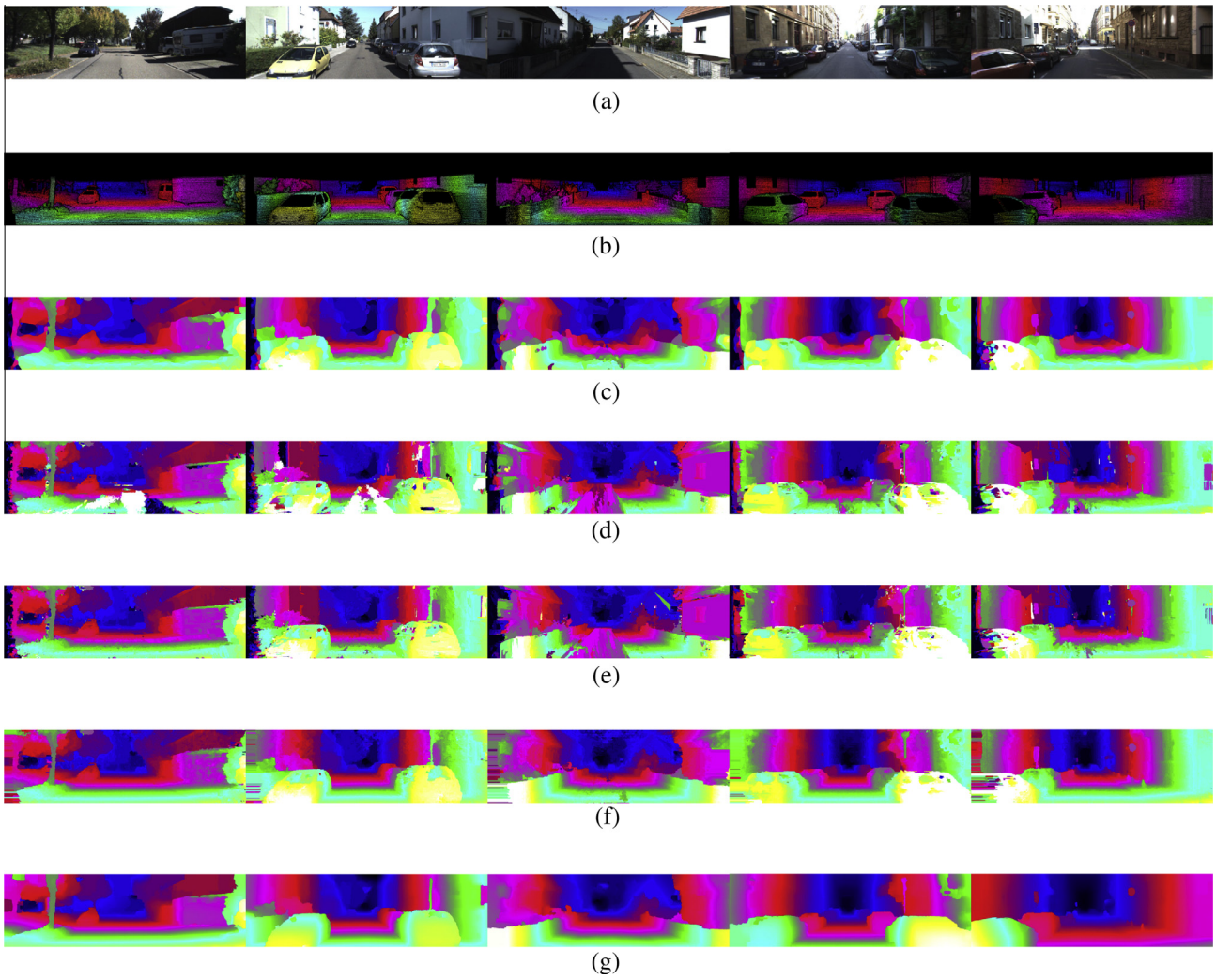

Fig. 8. Disparities results of different algorithms on KITTI dataset (frames #000038, #000051, #000054, #000063, #000070). (a) Input image, (b) ground truth, (c) GF [5], (d) CRF [16], (e) rSGM [55], (f) SPS [56], (g) proposed method.

Middlebury dataset, which are illustrated in Fig. 7. It can be found in Fig. 7 that even in the other stereo images, the proposed energy function can still outperform other functions.

Moreover, we further conduct the experiments on the KITTI training set for quantitative analysis. The results are shown in Table 1, where Out-All and Out-Noc represent the percentage of bad pixels in total and non-occluded regions, respectively. From Table 1, it can be found that the proposed energy function achieves the best performance with an error of 3.65% in non-occluded regions. Unless stated, the outlier threshold is 3 pixels. From Figs. 6, 7 and Table 1, several conclusions could be made: (1) Adding smoothness term to the data only term model help improve the accuracy; (2) Adding object constrain to the CRF model obviously decrease the error; (3) Our DCRF model has the best performance by combining the CRF model with CNN in the energy function.

5.2. Evaluation on the KITTI dataset

In this section, we implement the stereo framework introduced in Section 2. Experiments are conducted on the KITTI and Middlebury datasets. Recently, many researchers have been focusing on the stereo matching based on the MRF/CRF framework. Pal et al. [16] proposed the CRF model for stereo. Yamaguchi et al. [54] introduced the continuous MRF model for stereo matching. Moreover, as the development of artificial intelligence, the convolutional neural network is adopted for measuring the image similarity to obtain disparity estimation. Žbontar and LeCun [26] proposed the MC-CNN model to compute the matching cost. Chen et al. [27] presented a deep visual corresponding embedding model for cost measurement. Different form the above methods, our model contains the CNN term and implements the inference by RNN instead of the SGM method [55]. First, we carry out the

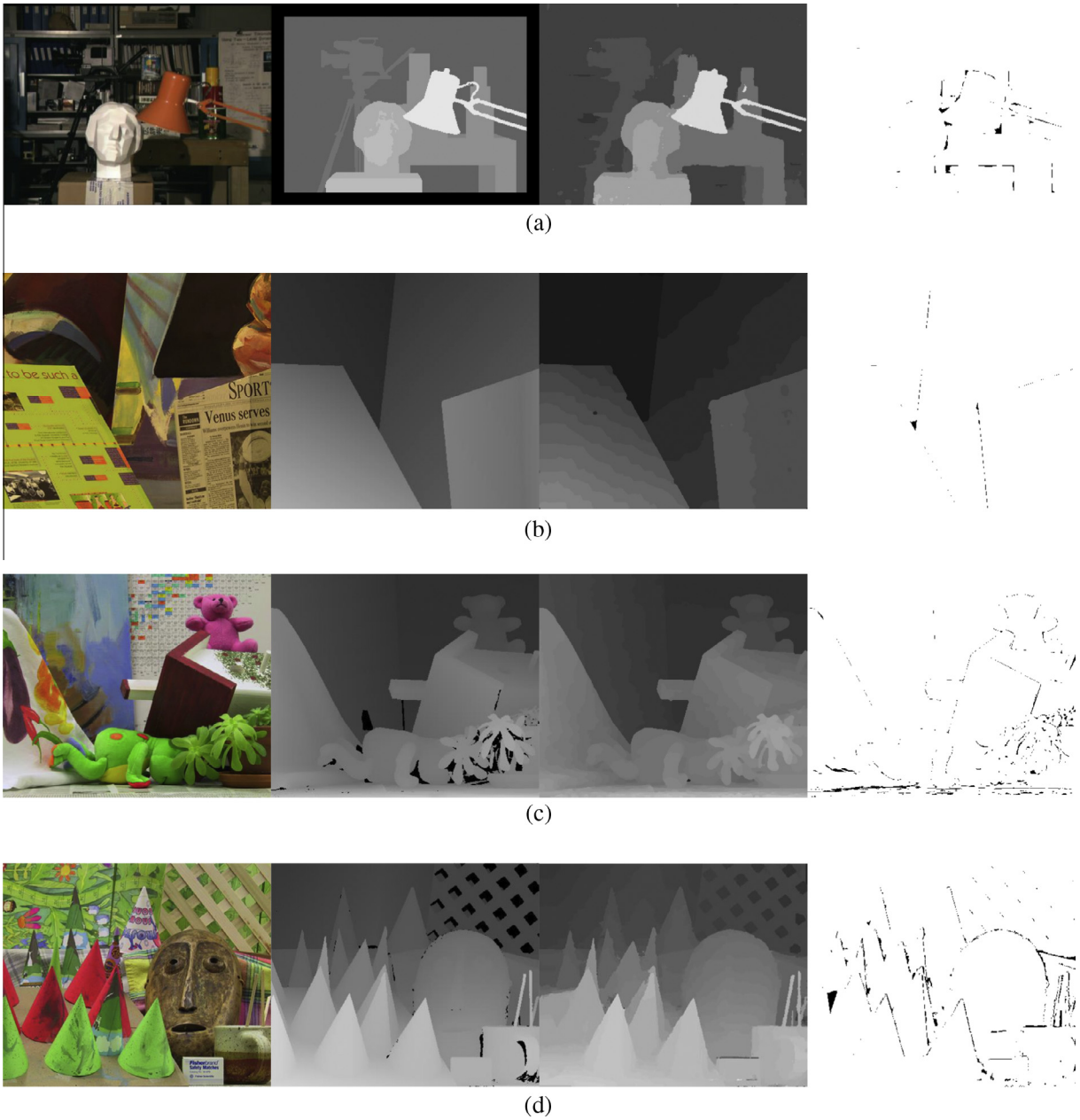

Fig. 9. Results of the proposed method on Middlebury dataset. (a) Tsukuba. (b) Venus. (c) Teddy. (d) Cones.

experiment on the KITTI dataset. The KITTI dataset contains 194 image pairs for training and 195 pairs for testing. These images are captured under real environments, which contain radiometric variations and noises. In this paper, comparisons are made with the CRF method [16], the SPS method [56] and the semi-global method rSGM [55]. Moreover, the GF-based method is also considered for a better evaluation. The testing results are reported in Table 2 and the visual comparisons are illustrated in Fig. 8.

In Table 2, four indicators are evaluated: Out-Noc, Out-All, Avg-Noc, Avg-All, where Avg-All and Avg-Noc represent average errors in all and non-occluded regions, respectively. From Table 2, it can be demonstrated that the DCRF-based method outperforms the MRF-based method PCBP, which shows the effectiveness of the combination of object knowledge and CNN in the energy function. The average error percentage of the proposed method decreases by 0.75% comparing with PCBP. Moreover, the proposed algorithm also produces more accurate results than the semi-global method rSGM and the local method GF, which further proves the better accuracy of global method for stereo matching. The average error percentage of the proposed method decreases by 1.74% comparing with rSGM and 8.36% comparing with GF. From Fig. 8 we can find that the disparity maps generated by the proposed method preserve the edges better and have less error pixels.

5.3. Evaluation on the Middlebury dataset

It is important to evaluate the proposed algorithm on different datasets to test the adaptability. We further evaluate the DCRF method on the Middlebury dataset, which consists of more than 30 stereo image pairs. At first, experiments are conducted on four stereo image pairs (i.e. Tsukuba, Venus, Cones, Teddy). The disparity results are shown in Fig. 9. Then, we extend the experiment to the Middlebury 2005 and 2006 dataset. The visual performances are illustrated in Fig. 10.

5.4. Post-processing evaluation

The post-processing step can obviously reduce the error, which is essential in almost all the stereo matching algorithms. To obtain a better understanding of the effectiveness of the post-processing steps, additional experiments are performed. The results are illustrated in Fig. 11.

Now, the performance of each post-processing step will be evaluated. The first step is the LRC check. Surprisingly, there appears a slight increment of error percentage after the first step due to the existence of noise. But it drops down quickly after following steps. After the four-direction propagation, the average error percentage is reduced by 0.81% and 1.26% in *nonocc* and *all* regions, respectively. This shows the effectiveness of the four-direction propagation in correcting outliers. The gradient domain guided image filtering step further reduces the error (e.g. 1.37% decrement in *all* regions and 0.6% decrement in *nonocc* region). In summary, significant improvements can be obtained after the multi-step refinement. Moreover, the performance of the post-processing approaches is checked on the training set of KITTI while other steps are the same. The results are reported in Table 3. *raw* is the raw disparity results. *pro* and *gif* represent the four-direction propagation and gradient domain guided image filtering, respectively.

5.5. Evaluation of design choices

To study the effectiveness of some design choices made above, additional experiments are performed on the KITTI training set. We first study the performance of the RNN parameter iteration. Fig. 12 shows how the iteration count T affects the disparity accuracy. From Fig. 12, it can be found that setting $T = 5$ produces an improvement in error with 1.05%, and setting $T = 10$ reduces the error by 2.28%. At last the error stabilizes after about 17 iterations. There is merely no improvement in accuracy when further increas-

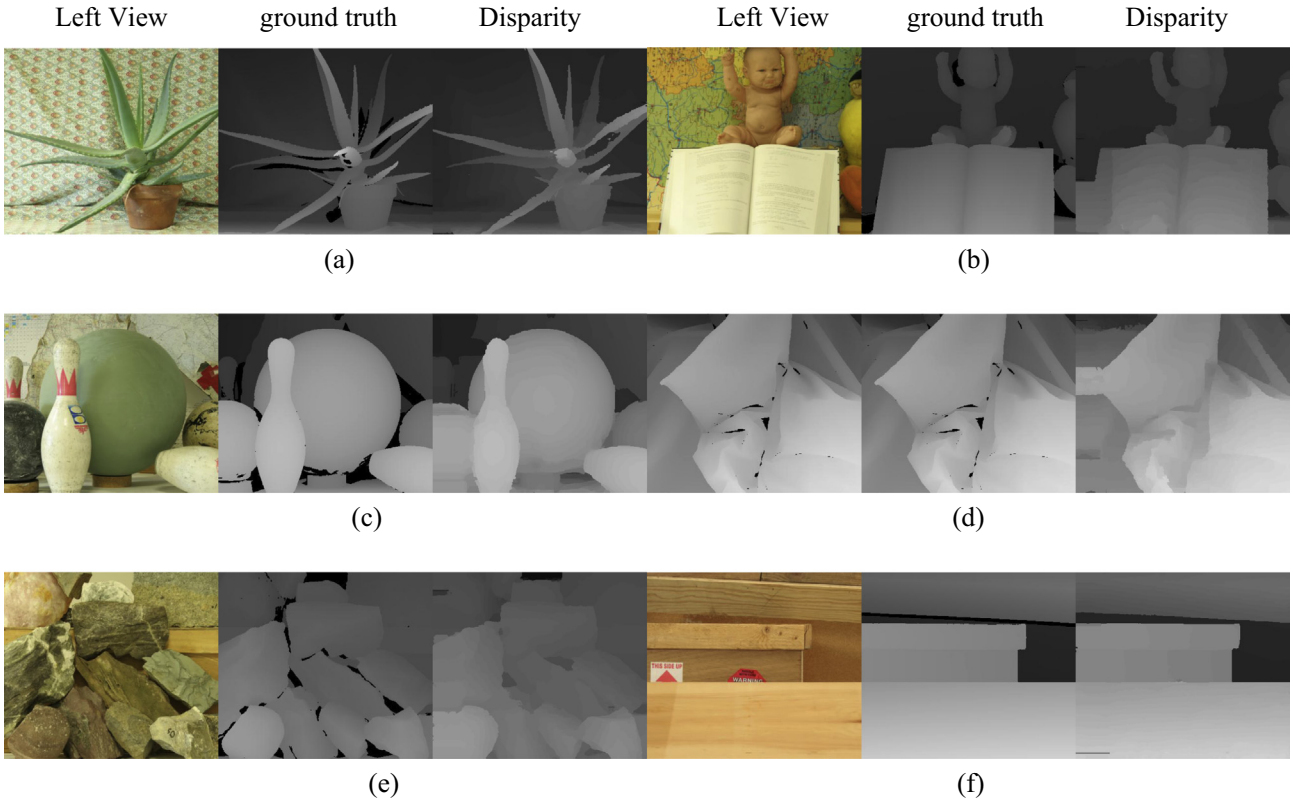

Fig. 10. The Middlebury 2005 and 2006 datasets. (a) Aloe. (b) Baby2. (c) Bowling2. (d) Cloth2. (e) Rocks2. (f) Wood2.

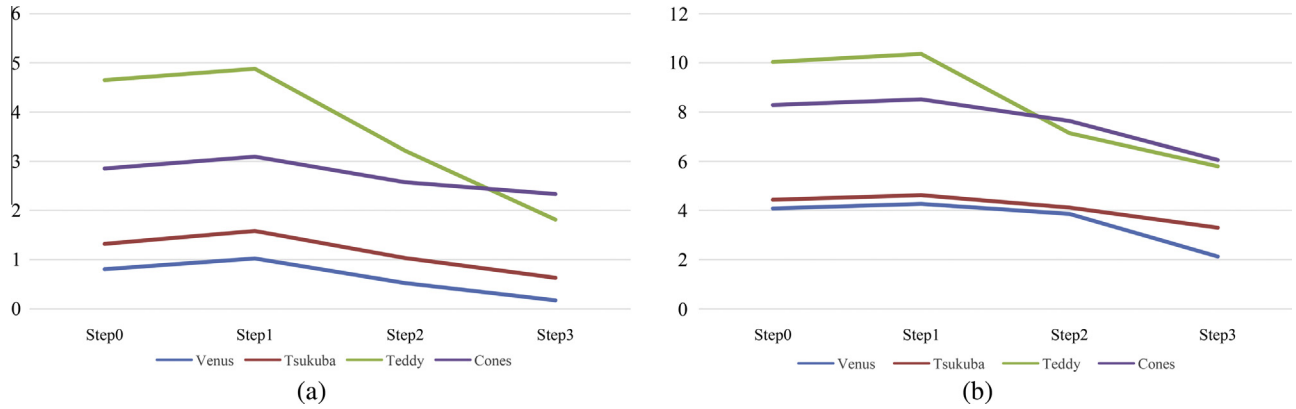

Fig. 11. Average error percentage after each post-processing step. (a) Error percentage in *nonocc* regions. (b) Error percentage in *all* regions. The operators from Step0 to Step3 are raw disparity maps, LRC check, four-direction propagation, gradient domain guided image filtering.

Table 3
Evaluation of different post-processing methods.

Algorithm	Average error percentage
raw	5.37
raw + LRC	5.28
raw + LRC + *pro*	4.35
raw + LRC + *gif*	4.16
raw + LRC + *gif* + *pro*	3.14

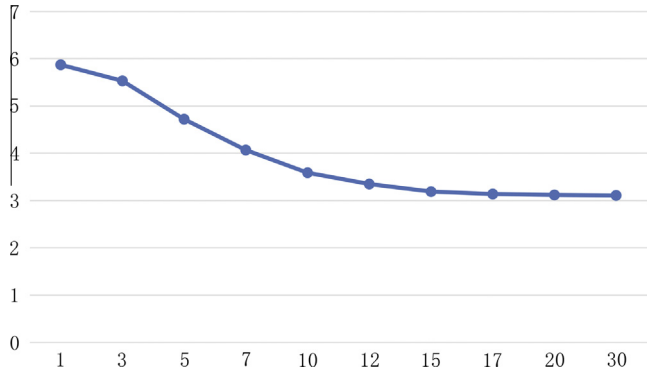

Fig. 12. Average error percentage in *nonocc* regions when varying the number of iterations.

ing the iteration number, which might be due to the vanishing gradient effect. Thus, the maximum number of iteration is set to 17 throughout all the experiments.

6. Conclusion

In this paper, a deep conditional random fields (DCRF) model is proposed, which extend the traditional CRF by incorporating the CNN into the energy function and formulating the inference of CRF into the RNN framework. The DCRF method outperforms the MRF/CRF-based method due to the combination of strength of CRF and CNN. Moreover, it also outperforms the semi-global method and local method. Experimental results show that the proposed algorithm can produce accurate results in both indoor and outdoor environments. Experiments on different datasets also show the adaptability and effectiveness of the DCRF-based method. In the future, we will extend the study in terms of the optimization of the computational complexity and application to more challenging scenes.

Acknowledgments

The authors would like to thank the National Natural Science Foundation of China (Grant No: 51521064), the Hangzhou Civic Significant Technological Innovation Project of China (No: 20131110A04), Hangzhou Civic Significant Technological Innovation Project of China (No: 20142013A56) for supporting this work.

References

[1] M.Z. Brown, D. Burschka, G.D. Hager, Advances in computational stereo, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 993–1008.
[2] L.E. Gurrieri, E. Dubois, Depth consistency and vertical disparities in stereoscopic panoramas, J. Electron. Imag. 23 (2014) 011004:1–011004:14.
[3] Q. Wang, Z. Yu, C. Rasmussen, J. Yu, Stereo vision–based depth of field rendering on a mobile device, J. Electron. Imag. 23 (2014) 1709–1717.
[4] Kuk-Jin Yoon, In So Kweon, Adaptive support-weight approach for correspondence search, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 650–656.
[5] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, M. Gelautz, Fast cost-volume filtering for visual correspondence and beyond, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 504–511.
[6] Q. Yang, A non-local cost aggregation method for stereo matching, in: IEEE Conference on Computer Vision & Pattern Recognition, 2012, pp. 1402–1409.
[7] Y. Zhan, Y. Gu, K. Huang, C. Zhang, K. Hu, Accurate image-guided stereo matching with efficient matching cost and disparity refinement, IEEE Trans. Circ. Syst. Video Technol. PP (2015) 1–14.
[8] M.G. Mozerov, J.V.d. Weijer, Accurate stereo matching by two-step energy minimization, IEEE Trans. Image Process. 24 (2015) 1153–1163.
[9] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, vol. 1, 1999, pp. 377–384.
[10] A.F. Bobick, S.S. Intille, Large occlusion stereo, Int. J. Comput. Vis. 33 (1999) 181–200.
[11] Q. Yang, Stereo matching using tree filtering, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 834–846.
[12] S. Birchfield, C. Tomasi, Multiway cut for stereo and motion with slanted surfaces, The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, vol. 1, 1999, pp. 489–495.
[13] Q. Yang, C. Engels, A. Akbarzadeh, Near real-time stereo for weakly-textured scenes, in: British Machine Vision Conference 2008, Leeds, September 2008.
[14] M. Bleyer, C. Rother, P. Kohli, Surface stereo with soft segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, 2010, pp. 1570–1577.
[15] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.
[16] C.J. Pal, J.J. Weinman, L.C. Tran, D. Scharstein, On learning conditional random fields for stereo, Int. J. Comput. Vis. 99 (2012) 319–337.
[17] C. Strecha, T. Tuytelaars, L.V. Gool, Dense matching of multiple wide-baseline views, Proceedings, Ninth IEEE International Conference on Computer Vision, 2003, vol. 2, 2003, pp. 1194–1201.
[18] S. Xu, F. Zhang, X. He, X. Shen, X. Zhang, PM-PM: patchmatch with potts model for object segmentation and stereo matching, IEEE Trans. Image Process. 24 (2015) 2182–2196.
[19] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, A comparative study of energy minimization methods for

Markov random fields with smoothness-based priors, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2007) 1068–1080.
[20] C. Yan, Y. Zhang, F. Dai, X. Wang, L. Li, Q. Dai, Parallel deblocking filter for HEVC on many-core processor, Electron. Lett. 50 (2014) 367–368.
[21] C. Yan, Y. Zhang, J. Xu, F. Dai, L. Li, Q. Dai, F. Wu, A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors, IEEE Signal Process. Lett. 21 (2014) 573–576.
[22] C. Yan, Y. Zhang, F. Dai, L. Li, Highly parallel framework for HEVC motion estimation on many-core platform, in: Data Compression Conference (DCC), 2013, 2013, pp. 63–72.
[23] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, F. Wu, Efficient parallel framework for HEVC motion estimation on many-core processors, IEEE Trans. Circ. Syst. Video Technol. 24 (2014) 2077–2089.
[24] L. Zhang, S.M. Seitz, Parameter estimation for MRF stereo, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, 2005, pp. 288–295.
[25] H. Trinh, D.A. Mcallester, Unsupervised learning of stereo vision with monocular depth cues, in: British Machine Vision Conference, BMVC 2009, London, UK, September 7–10, 2009, Proceedings, 2009.
[26] J. Žbontar, Y. Lecun, Stereo matching by training a convolutional neural network to compare image patches, Comput. Sci. 17 (2015) 2287–2318.
[27] Z. Chen, X. Sun, L. Wang, Y. Yu, C. Huang, A deep visual correspondence embedding model for stereo matching costs, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 972–980.
[28] Y. Taguchi, B. Wilburn, C.L. Zitnick, Stereo reconstruction with mixed pixels using adaptive over-segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, CVPR 2008, 2008, pp. 1–8.
[29] W. Zeng-Fu, Z. Zhi-Gang, A region based stereo matching algorithm using cooperative optimization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, CVPR 2008, 2008, pp. 1–8.
[30] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, S. Sinha, Object stereo—joint stereo matching and object segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, 2011, pp. 3081–3088.
[31] J.S. Yedidia, W.T. Freeman, Y. Weiss, Understanding belief propagation and its generalizations, Explor. Artif. Intell. New Millenium 54 (2003) 276–286.
[32] A.T. Ihler, D.A. Mcallester, Particle belief propagation, J. Mach. Learn. Res. 5 (2009) 256–263.
[33] T. Hazan, A. Shashua, Norm-product belief propagation: primal-dual message-passing for approximate inference, IEEE Trans. Inform. Theory 56 (2010) 6294–6316.
[34] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Comput. Sci. (2013) 580–587.
[35] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Comput. Sci. 79 (2014) 1337–1342.
[36] J. Tompson, A. Jain, Y. Lecun, C. Bregler, Joint training of a convolutional network and a graphical model for human pose estimation, Eprint Arxiv, 2014, pp. 1799–1807.
[37] K. Yamaguchi, D. McAllester, R. Urtasun, Robust monocular epipolar flow estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, 2013, pp. 1862–1869.
[38] L. Ladický, C. Russell, P. Kohli, P.H. Torr, Associative hierarchical random fields, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2013) 1056–1077.
[39] Z.Y. Zhang, Recurrent neural network, Scholarpedia 8 (2013) 1537–1541.
[40] F. Besse, C. Rother, A. Fitzgibbon, J. Kautz, PMBP: patchmatch belief propagation for correspondence field estimation, Int. J. Comput. Vis. 110 (2014) 2–13.
[41] Z. Li, J. Zheng, Z. Zhu, W. Yao, S. Wu, Weighted guided image filtering, IEEE Trans. Image Process. 24 (2015) 120–129.
[42] X. Mei, X. Sun, M. Zhou, S. Jiao, On building an accurate stereo matching system on graphics hardware, in: IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 2011, pp. 467–474.
[43] P. Kohli, L.U. Ladický, P.H.S. Torr, Robust higher order potentials for enforcing label consistency, in: Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[44] S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4353–4361.
[45] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inform. Process. Syst. 25 (2012) 2012.
[46] F. Liu, C. Shen, G. Lin, Deep convolutional neural fields for depth estimation from a single image, in: Computer Vision and Pattern Recognition, 2014.
[47] P. Krähenbühl, V. Koltun, Efficient inference in fully connected CRFs with Gaussian edge potentials, in: Advances in Neural Information Processing Systems, 2015, pp. 109–117.
[48] C. Farabet, C. Couprie, L. Najman, Y. Lecun, Learning hierarchical features for scene labeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1915–1929.
[49] A. Klaus, M. Sormann, K. Karner, Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure, Proceedings of the 18th International Conference on Pattern Recognition, vol. 03, IEEE Computer Society, 2006, pp. 15–18.
[50] Q. Yang, L. Wang, R. Yang, H. Stewénius, D. Nistér, Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2008) 492–504.
[51] F. Kou, W. Chen, C. Wen, Z. Li, Gradient domain guided image filtering, IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. 24 (2015) 4528–4539.
[52] Q. Yang, P. Ji, D. Li, S. Yao, M. Zhang, Fast stereo matching using adaptive guided filtering, Image Vis. Comput. 32 (2014) 202–211.
[53] V. Turchenko, A. Luczak, Caffe: convolutional architecture for fast feature embedding. Eprint Arxiv, 2014, pp. 675–678.
[54] K. Yamaguchi, T. Hazan, D. Mcallester, R. Urtasun, Continuous Markov random fields for robust stereo estimation, Lect. Notes Comput. Sci. 7576 (2012) 45–58.
[55] R. Spangenberg, T. Langner, S. Adfeldt, R. Rojas, Large scale semi-global matching on the CPU, in: Intelligent Vehicles Symposium Proceedings, 2014, pp. 195–201.
[56] K. Yamaguchi, D. McAllester, R. Urtasun, Efficient joint segmentation, occlusion labeling, stereo and flow estimation, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V, Springer International Publishing, Cham, 2014, pp. 756–771.
[57] A. Chakrabarti, Y. Xiong, S.J. Gortler, T. Zickler, Low-level vision by consensus in a spatial hierarchy of regions, Eprint Arxiv, 2014.
[58] D. Wei, C. Liu, W.T. Freeman, A data-driven regularization model for stereo and flow, in: International Conference on 3D Vision, 2014, pp. 277–284.
[59] J.H. Cho, M. Humenberger, Fast PatchMatch stereo matching using cross-scale cost fusion for automotive applications, in: Intelligent Vehicles Symposium, 2015.
[60] N. Einecke, J. Eggert, A multi-block-matching approach for stereo, in: Intelligent Vehicles Symposium, 2015.
[61] S. Lee, J.H. Lee, J. Lim, I.H. Suh, Robust stereo matching using adaptive random walk with restart algorithm, Image Vis. Comput. 37 (2015) 1–11.