Occlusion Boundary Detection via Deep Exploration of Context

Huan Fu, Chaohui Wang, Dacheng Tao, Michael J Black

Occlusion Boundary Detection via Deep Exploration of Context

Huan Fu[1] Chaohui Wang[2] Dacheng Tao[1] Michael J. Black[3]
[1]QCIS and FEIT, University of Technology Sydney
[2]Université Paris-Est, LIGM - CNRS UMR 8049
[3]Max Planck Institute for Intelligent Systems, Tübingen, Germany
Huan.Fu@student.uts.edu.au chaohui.wang@u-pem.fr dacheng.tao@uts.edu.au black@tuebingen.mpg.de

Abstract

*Occlusion boundaries contain rich perceptual information about the underlying scene structure. They also provide important cues in many visual perception tasks such as scene understanding, object recognition, and segmentation. In this paper, we improve occlusion boundary detection via enhanced exploration of contextual information (*e.g.*, local structural boundary patterns, observations from surrounding regions, and temporal context), and in doing so develop a novel approach based on convolutional neural networks (CNNs) and conditional random fields (CRFs). Experimental results demonstrate that our detector significantly outperforms the state-of-the-art (*e.g.*, improving the F-measure from 0.62 to 0.71 on the commonly used CMU benchmark). Last but not least, we empirically assess the roles of several important components of the proposed detector, so as to validate the rationale behind this approach.*

1. Introduction

Occlusions are ubiquitous in 2D images of natural scenes (Fig. 1). They are introduced in the 3D-to-2D projection process during the image formation, due to the overlapping of the 2D extents (in the image plane) of 3D components/surfaces. In this paper, we focus on the problem of detecting *occlusion boundaries*, each of which separates two 2D regions projected from two parts of scene surfaces that overlap locally in either of those regions.

Occlusion boundary detection is of interest in computer vision, image analysis, and other related fields (*e.g.*, [14, 15, 34, 45]). Indeed, occlusions constitute an obstacle to designing rigorous models and efficient algorithms in computer vision and image analysis. Besides the lack of information on invisible scene components, another main reason is that occlusions invalidate the assumption that two neighboring pixels in a 2D image correspond to two adjacent points lying on a common part of a 3D surface. Nevertheless, this invalid assumption is often made, either explicitly or implicitly, in existing methods (*e.g.*, the use of smoothness priors for aggregating spatial information in the 2D image). The localization of occlusion boundaries would, therefore, be very useful for overcoming this limitation and improving the solution in these tasks. Furthermore, since occlusion boundaries separate visible scene components from locally occluded components and usually correspond to an abrupt change in depth (along the line of view), these boundaries contain rich perceptual information about the underlying 3D scene structure, the exploitation of which would be beneficial in various visual perception applications. For example, occlusion boundaries can serve as important cues for object discovery and segmentation (*e.g.*, [3, 14]), since an object is generally delimited from its environment by the isolation of its 3D surface. Indeed, psychologists have long studied their importance in human visual perception (*e.g.*, Gibson, Biederman) [5, 17].

Figure 1. Illustration of ubiquitous occlusions and local occlusion patterns (source image from [38]).

Despite the considerable number of studies on occlusion boundary detection (*e.g.*, [18, 21, 35, 36, 40, 44]), the state-of-the-art is still unsatisfactory. We believe that one main reason for this is that contextual information has not been sufficiently explored in an efficient way. In fact, numerous previous studies developed their approaches based on structural modeling tools, *e.g.*, *Markov/Conditional Ran-*

dom Fields (MRFs/CRFs) [8, 48], to exploiting contextual information and demonstrated the importance of this information in solving various computer vision and image analysis problems (*e.g.*, [11, 23, 31, 33, 46, 47, 53, 54]). This has motivated us to better explore contextual information in detecting occlusion boundaries.

In this work, we are interested in exploring three main types of contextual information that are useful for occlusion boundary detection: (i) local contextual correlations in pixel labeling[1] (*e.g.* [16, 18, 21, 35, 36, 40, 44]); (ii) contextual correlations between the labeling of pixels (*e.g.*, patch) and the observations from the surrounding area of the region (*e.g.*, [16, 21]); and (iii) temporal contextual information contained in video sequences (*e.g.*, [35,40]). Moreover, we aim to jointly model these three types of information so as to better explore them and boost occlusion boundary detection performance. To this end, we finally propose a novel approach for occlusion boundary detection based on convolutional neural networks (CNNs) [4] and CRFs [8, 48], two of the most powerful and successful modeling tools in computer vision and related fields.

More specifically, in order to better explore type (i) and (ii) contextual information, our CNN model considers a relatively big image patch "L" as input, performs reasoning on it, and outputs the state of a relatively small patch "S" with the same center with "L" (referred to as *L2S*). It provides not only a probabilistic labeling map on "S", but also deep features that aggregate the high-level contextual information on "L". These are then fed into our CRF model to globally infer the occlusion boundaries in the whole image. For exploring type (iii) contextual information, we consider the scenario in which a video sequence is the input data (similar to many existing works *e.g.*, [18, 35, 36, 40, 44])[2], and two simple optical-flow-based motion features are exploited to efficiently capture temporal contextual information.

Experimental results demonstrate that the proposed detector significantly outperforms the state-of-the-art, *e.g.*, by improving the F-measure from 0.62 to 0.71 on the commonly used CMU benchmark [40] (see Fig. 2 for the precision-recall curves and Tab. 1 for the F-measures). Last but not least, we empirically demonstrate the importance of spatial and temporal contextual information in occlusion boundary detection, and compare several CNN-based alternative methods to illustrate that L2S provides more robust and discriminative deep occlusion features than those variants. These validate the underlying rationale of our approach, which would also be helpful for addressing other visual perception tasks.

[1]Like most existing methods, we formulate the problem by endowing each pixel with a binary variable denoting whether the pixel is on occlusion boundaries.

[2]It should be noted that our method can also be applied directly to the scenario where an individual 2D image is the input (see Tab. 3).

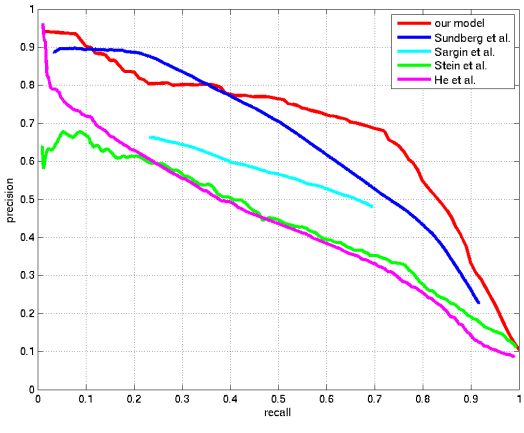

Figure 2. **Precision-recall curves.** Based on the commonly used CMU benchmark [40], we compare our results with those of Sundberg et al. [44], Sargin et al. [36], Stein et al. [40] and He et al. [18].

Related Work

Occlusion boundary estimation has attracted extensive attention in computer vision over the past few years. Several methods have been proposed to detect occlusion boundaries in a single image. For example, Saxena *et al*. [37] learn an MRF to capture 3D scene structure and depth information from single images. Hoiem *et al*. [21] demonstrate the importance of 2D perceptual cues and 3D surface and depth cues for occlusion boundary detection, and compute these geometric contexts to reason about occlusions within their CRF model.

Due to the fact that occlusion boundary detection from a single 2D image is ambiguous, many applications consider videos or image sequences as inputs and extend occlusion boundary detection to the temporal dimension. Apostoloff and Fitzgibbon [1] observe that the T-junction is a particularly strong occlusion indicator, and thus learn a relevance vector machine (RVM) T-junction classifier on spatiotemporal patches and fuse Canny edges and T-junctions to detect occlusion edges in the spatial domain. Feldman and Weinshall [14] define the average of the second moment matrix around a pixel as a gradient structure tensor by regarding the video sequence as a spatiotemporal intensity function, and demonstrate that the smallest eigenvalue of this tensor is the occlusion indicator. Stein and Hebert [40] exploit subtle relative motion cues present at occlusion boundaries during a sequence of frames and develop a global boundary model that combines these motion cues and standard appearance cues based on an initial edge detector [30]. Black and Fleet [7] represent occlusion boundaries via a generative model that explicitly encodes the orientation of the boundary, the velocities on either side, the motion of the occluding edge over time, and the appear-

ance/disappearance of pixels at the boundary. Based on this model, the motion of occlusion boundaries is predicted and thus information over time is integrated. Besides, both motion boundaries and image boundaries are combined within an MRF framework in [6] to better reason about the occlusion structure in the scene over time.

Although some aforementioned methods attempt to develop discriminative occlusion features on a spatiotemporal volume, recent works have shown that directly using flow-based occlusion features as the temporal information is more convenient and efficient. To name a few, Sargin *et al*. [36] introduce a probabilistic cost function to generate a spatiotemporal lattice across multiple frames to produce a factor graph. Boundary feature channels are then learned using this factor graph, by taking some independent flow-based occlusion feature channels into account. He and Yuille [18] argue that image depth discontinuities often occur at occlusion boundaries and estimate the pseudo-depth using the singular value decomposition (SVD) technique from motion flow as a cue for their occlusion detector. Sunberg *et al*. [44] recompute occlusion motion flows on each edge fragment at region boundaries from the initial optical flow [10], based on the observation that an occlusion boundary can be handled by comparing the difference in optical flow in regions on either side. Reporting that local patch features are unable to handle highly variable appearances or intra-object local motion, Raza *et al*. [35] estimate temporally consistent occlusion boundaries via an MRF model whose potentials are learned by random forests using global occlusion motion features and a high-level geometric layout on segmentation boundaries.

Regarding contextual information, it has already proved to play an important role in many computer vision tasks such as object detection, localization, and recognition [11, 29, 31, 47]. Recently, context modeling has also been introduced to boundary detection. Dollár and Zitnick [12] adopt random decision forests [25] to capture structural information of local patch edges. Weinzaepfel *et al*. [51] extend [12] to video datasets and exploit temporal information and static image cues to learn correlations between motion edges within local patches (edges between motion objects).

Previous studies have suggested that the brain encodes contextual information and biologically inspired deep CNNs have been shown to be powerful for feature extraction and description [19, 26], which have motivated us to learn the internal correlation of an occlusion boundary in local patches using the CNN framework. Patch-level CNNs have been widely used in a variety of computer vision tasks, with excellent progress made over recent years. For example, Fan *et al*. [13] combine local image patches and a holistic view in a CNN framework to learn contextual information for human pose estimation. Wang *et al*. [49] exploit physical constraints in local patches using a CNN-based model or surface normal estimation. Sun *et al*. [43] and Li *et al*. [28] learn convolutional features from multiple local regions for facial trait recognition. Besides, Sun *et al*. [42] formulate an MRF model to remove non-uniform motion blur using the patch-level probabilistic motion blur distribution by CNNs. Motivated by nearest neighbor relationships within a local patch, Ganin and Lempitsky [16] detect edges by learning a 4×4 label feature vector for each patch and matching against a sample CNN output dictionary corresponding to training patches with known annotation. Shen *et al*. [39] make use of the structural information of object contours in contour detection, by classifying image patches into different boundary types and accordingly defining a special loss function for training a CNN.

2. The Proposed Occlusion Boundary Detector

Firstly, in order to better characterize local contextual correlations in pixel labeling and contextual correlations between regional pixel labeling and surrounding observations, we: (i) consider each individual pixel patch as the unit of interest, and (ii) adopt a CNN to learn and predict a patch's occlusion boundary map based on the observation of a larger patch of pixels with the same center. Secondly, we efficiently explore and encode temporal contextual information within the whole framework by adopting effective motion features in the CNN. Finally, we use a CRF model to integrate patch-based occlusion boundary maps and soft contextual correlations between neighboring pixels to achieve occlusion boundary estimation for the entire image. Each part is described below.

2.1. Patch-based Labeling using CNNs

We are interested in modeling and predicting labeling of a patch of pixels based on the observation of a larger patch with the same center via a structured learning/prediction process. Mathematically, given the K-channel observed data on an $N \times N$ patch centered at pixel c, denoted $\mathbf{x}_c \in \mathbb{R}^{K*N^2}$, we aim to obtain the weighted occlusion boundary map $\mathbf{y}_c \in \mathbb{R}^{M^2}$ on an $M \times M$ ($M < N$) patch that is also centered at pixel c, which is achieved via our structured CNN convolutional neural network illustrated in Fig. 3. Below we first briefly describe the architecture of our structured CNN and then discuss the initial input features/cues used for occlusion boundary detection.

2.1.1 CNN Architecture

We train a CNN using a cross entropy loss function to predict the probability distribution in a small 7×7 patch from a large 27×27 image patch (*i.e.*, $M = 7$ and $N = 27$ in our experiments). The overall CNN architecture is shown in Fig. 3. The input of our CNN consists of 3 static color channels and 2 temporal channels

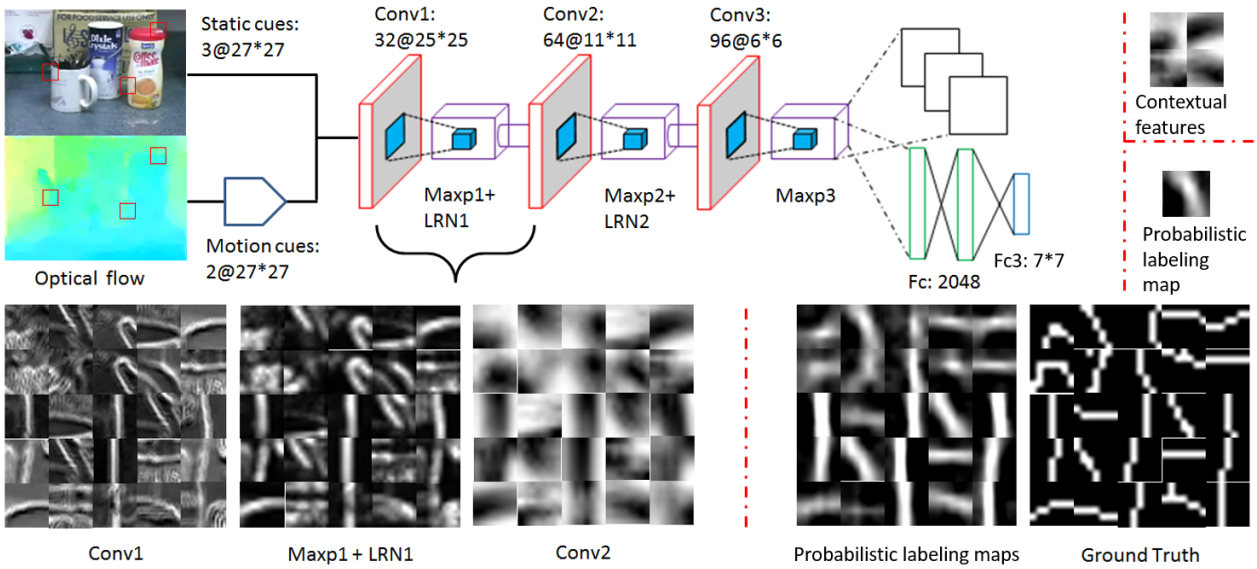

Figure 3. **Illustration of the CNN architecture and the output of several layers.** For a 27×27 patch of the given input sequence, we first extract 5 (3 static + 2 motion) initial feature maps, which is the input of the CNN. The output of $Maxp3$ layer corresponds to deep features that aggregate the high-level contextual information (referred to as *deep contextual features*). $fc3$ layer outputs a probabilistic labeling map on a 7×7 patch.

of size 27×27 (detailed in Sec. 2.1.2). The CNN structure can be described by the size of the feature map at each layer as follows: $conv1$ (32@25*25) $\rightarrow$ $maxp1$ $\rightarrow$ $LRN1$ $\rightarrow$ $conv2$ (64@11*11) $\rightarrow$ $maxp2$ $\rightarrow$ $LRN2$ $\rightarrow$ $conv3$ (96@6*6) $\rightarrow$ $maxp3$ $\rightarrow$ $fc1$ (2048) $\rightarrow$ $dropout1$ $\rightarrow$ $fc2$ (2048) $\rightarrow$ $dropout2$ $\rightarrow$ $fc3$ (49), which corresponds to a probabilistic labeling map of size 7×7. Here, $conv$, $maxp$, LRN, fc, and $dropout$ denote the convolutional layer, max pooling layer, local response normalization layer, fully-connected layer, and dropout layer, respectively. The LRN scheme implements a form of lateral inhibition, encouraging competition for large activations in the neuronal output [9,32]. The dropout layer is used to prevent units from co-adapting too much when training a large neural network [20].

In our CNN architecture, the rectified linear units $(ReLUs)$ non-linear active function, $f(x) = max(0, x)$, is followed by all $conv$ and fc layers except $fc3$. A sigmoid function is applied to $fc3$ to obtain a probabilistic labeling map; and accordingly the cross entropy loss function is adopted for the training process. Furthermore, the output of $Maxp3$ provides learned deep features that aggregate the high-level contextual information (referred to as *deep contextual features*), which are then used in the CRF model (see Sec. 2.2) to globally reason about occlusion boundaries.

2.1.2 Initial Features for Occlusion Reasoning

Many previous occlusion boundary detection methods have attempted to extract various specific features that characterize occlusions in raw images such as T-junctions, relative depths, and other useful 3D scene properties [18, 21, 35]. However, accurate automatic extraction of such features is also challenging. To avoid these difficulties, we aim to perform occlusion reasoning by using simple but effective initial features/cues. To this end, we first convert an RGB image to Lab space and consider the gradient magnitude of the Lab maps as three feature maps for the CNN model. In addition, we include two optical-flow-based motion features to efficiently encode temporal contextual information in video sequences and further improve detection performance. Finally, the input of our CNN model consists of 5 (3 static + 2 motion) feature maps. The two motion features are detailed below.

- **Motion Feature 1** The first occlusion motion feature OMF_1 aims to capture optical flow discontinuity which suggests occlusion boundaries. We use $f_{t,t+t_0}$ $(t_0 \in \mathcal{N}^*)$ to denote the optical flow map from frame t to frame $t + t_0$ ($t_0 = 5$ in the experiments). To capture the discontinuity of $f_{t,t+t_0}$, we compute the unoriented gradient magnitude $GF_{t,t+t_0}$:

$$GF_{t,t+t_0} = |\nabla f_{t,t+t_0}| \tag{1}$$

Since both forward flow $f_{t,t+t_0}$ and backward flow $f_{t,t-t_0}$ provide motion information from frame t, in order to achieve robustness, we compute $GF_{t,t-t_0}$ similarly together with $GF_{t,t+t_0}$ and consider their geometric mean as one occlusion motion feature OMF_1:

$$OMF_1 = \sqrt{GF_{t,t+t_0} * GF_{t,t-t_0}} \tag{2}$$

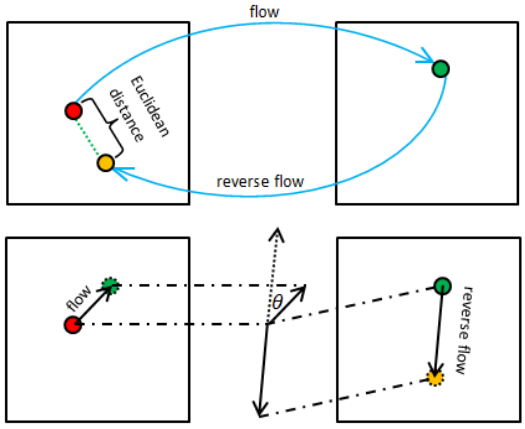

Figure 4. **Illustration of flow inconsistencies.**

- **Motion Feature 2** The second occlusion motion feature OMF_2 models the fact that the consistency of the flow $f_{t,t+t_0}$ and reverse flow $f_{t+t_0,t}$ is not satisfied when occlusion occurs [22]. We measure these inconsistencies using both location and angle, illustrated in Fig. 4. Let f_l and f'_l denote the flow values at location l in the forward and reverse flow maps $f_{t,t+t_0}$ and $f_{t+t_0,t}$, respectively. If a point located at l in frame t is visible at t and $t+t_0$, its correspondence in frame $t+t_0$ should be located at $l+f_l$ and should return to its start position l after being transported to frame t by the reverse flow f'_{l+f_l}. And with respect to angle, flow f_l and reverse flow f'_{l+f_l} should be π apart if consistent. Hence, we measure these inconsistencies as follow:

$$\Gamma_l = |f_l + f'_{l+f_l}| \tag{3}$$

$$\Lambda_l = \begin{cases} 0 & P|Q \\ arccos\{\frac{-f_l \cdot f'_{l+f_l}}{|f_l||f'_{l+f_l}|}\} & \text{others} \end{cases} \tag{4}$$

where P and Q represent $|f_l| < \delta$ and $|f'_{l+f_l}| < \delta$, respectively, and are used to filter out the likely static pixels and prevent the denominator of the formulation above from being 0 ($\delta = 0.01 * \max_l(|f_l|)$ for each frame in the experiments). Since both Γ and Λ describe inconsistent properties when occlusions occur, we combine them to obtain our inconsistency descriptor $IC_{t,t+t_0}$, via a Gaussian smoothness process:

$$IC_{t,t+t_0}(l) = \sum_{l^*} \sigma(d - |l^* - l|) e^{-\frac{|l^*-l|^2}{2}} \sqrt{\Gamma_{l^*} \Lambda_{l^*}} \tag{5}$$

where $\sigma(x) = 1$ when $x \geq 0$ and 0 otherwise, and $d = 2$ in the experiments. Similar to OMF_1, OMF_2 also takes the backward flow into consideration and is defined as:

$$OMF_2 = \sqrt{IC_{t,t+t_0} * IC_{t,t-t_0}} \tag{6}$$

2.2. Image-level Reasoning via CRFs

We then adopt CRFs to efficiently integrate patch-based occlusion boundary maps and soft contextual correlations between neighboring pixels, so as to achieve global occlusion boundary estimation for the entire image. Here, we consider the most common pairwise CRF with 4-neighborhood system used in computer vision and image analysis[3]. In the CRF model, the nodes correspond to the pixel lattice and the edges to pairs of neighboring nodes. Let $\mathcal{V}$ and $\mathcal{E}$ denote the node set and the edge set, respectively. The CRF energy is defined as:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{\{i,j\} \in \mathcal{E}} \theta_{ij}(x_{ij}) \tag{7}$$

Unary potentials $(\theta_i(\cdot))_{i \in \mathcal{V}}$ are used to encode the data likelihood on individual pixels based on the patch-based probabilistic labeling maps provided by the CNN presented in Sec. 2.1, by defining $\theta_i(\cdot)$ as the negative logarithm of the average probability $\bar{p}_i(\cdot)$ over all output patches that cover the pixel i:

$$\theta_i(x_i) = -\log \bar{p}_i(x_i) \tag{8}$$

Let R_i denote the deep contextual features of the local patch centered at pixel i provided by $maxp3$ layer of our CNN, and the l^2 norm between R_i and R_j is measured to capture the dissimilarity between neighboring pixels i and j. To penalize different labels between neighboring pixels, the *Pairwise potentials* $(\theta_{ij}(\cdot))_{\{i,j\} \in \mathcal{E}}$ between pairs of nodes are defined as:

$$\theta_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ w \cdot \exp\{- \parallel R_i - R_j \parallel\} & \text{otherwise} \end{cases} \tag{9}$$

where w is a weight coefficient balancing the importance of the unary and pairwise terms ($w = 2.1$ in the experiments).

2.3. Implementation Details

We adopt the region detector provided in [2], which produces a large number of small regions, so as to preserve nearly all types of boundaries, including occlusion boundaries. The occlusion boundary detection boils down to a binary classification problem which determines whether the boundary between regions is or is not an occlusion boundary, which is the same setting as many previous work (*e.g.*,

[3]The whole method is modular with respect to the choice of CRFs. A main reason to adopt the pairwise CRF with 4-neighborhood system in the experiments, instead of more sophisticated CRFs, is to demonstrate more clearly the effectiveness of the whole method.

[21, 35, 40, 44]). In order to address the ground truth labeling bias (*e.g.*, the original set of boundaries created by [2] are often 1 or 2 pixels away from the corresponding ground truth boundaries drawn by hand [40]), we consider all pixels within 2 pixels of the boundaries obtained by [2] to produce image patches[4]. To balance the number of positive patches (patches containing an occlusion boundary curve) and negative patches during training, we randomly sample 100,000 training patches in a 1:1 ratio.

The 3 image and 2 motion cue channels are the CNN input to learn internal correlations around occlusion boundaries and predict probabilistic labeling maps and extract deep contextual features. See Sec. 2.1 for the motion cues computation and structured CNN framework. Our structured CNN model is built based on *Caffe* [24], developed by the Berkeley Vision And Learning Center (BVLC) and community contributors. The CRF model is then constructed to globally estimate occlusion boundaries for each image using the probabilistic labeling maps and the deep contextual features provided by the learned CNN model. Regarding the CRF inference, many powerful off-the-shelf algorithms can be directly applied to solve the CRF model [48]. We simply used sum-product loopy belief propagation [52] to estimate approximate-marginal probabilities of all nodes/pixels via message passing over the graph, so as to get a probabilistic boundary labeling map on the entire image and directly compare with previous methods using the same quality metric, *i.e.*, F-measure.

In the final step, we apply the method in Arbelaez *et al*. [2] to remove isolated pixels and connect disconnected short lines that might belong to a long boundary in our probabilistic occlusion boundary map η. This produces contour boundary map Ω. We then combine η and Ω by learning a weight factor α using SVM to get obtain our final occlusion boundary detector ξ:

$$\xi = \alpha * \eta + (1 - \alpha) * \Omega \tag{10}$$

In this paper, the final α is 0.65.

3. Experimental Results

Following most previous work on occlusion boundary detection, such as [18, 27, 36, 40, 44], we evaluate the performance of the proposed detector on the CMU benchmark [40], and perform quantitative comparison with previous work using the precision (Pre) $vs.$ recall (Rec) curve and F-measure (F-measure $= \frac{2*Pre*Rec}{Pre+Rec}$) as quality measures. In addition, we also perform quantitative evaluation on the datasets provided by [44] and [35] and compare with their methods. In the following, we will first exhibit the obtained qualitative and quantitative results, and then empirically analyze the importance of several major components via the comparison between the corresponding variants of the proposed detector.

Algorithm	F-measure
Stein et al. [40]	0.48
Sargin et al. [36]	0.57
He et al. [18]	0.47
Sundberg et al. [44]	0.62
Leordeanu et al. [27]	0.62
Our detector $(probability)$	**0.71**
Our detector $(hard)$	**0.63**

Table 1. F-measure values (the average of maximal F-measure values over the whole benchmark) obtained by the considered methods on the CMU benchmark.

Algorithm	F-measure	
	Dataset [44]	Dataset [35]
Sundberg et al. [44]	0.56	N/A
Raza et al. [35]	N/A	0.60
Our detector	**0.59**	**0.63**

Table 2. Comparison with [44] and [35] on their datasets.

Qualitative and Quantitative Results A representative set of qualitative results on the CMU benchmark are exhibited in Fig. 5. The quantitative comparison with several important previous methods [18, 27, 36, 40, 44] is shown in Tab. 1 and Fig. 2, where the quantitative results of the previous methods are taken from their papers. Specifically, the $Precision$ $vs.$ $Recall$ curves of different methods are shown in Fig. 2, which indicates that our detector significantly outperforms previous methods, especially withn a recall interval of $[0.6, 0.9]$.

The evaluation based on F-measure is shown in Tab. 1, which also demonstrates a significant improvement over previous methods. Following previous work, we range the threshold on the probabilistic occlusion boundary map obtained for each image and compute the average of the maximal F-measure (*AMF*) across the whole dataset, and report the obtained results in Tab. 1. Moreover, considering that: (i) the optimal threshold that leads to the maximal F-measure of the occlusion boundary map generally varies between input test images, (ii) hard boundary labeling results (*i.e.*, each pixel is labeled either 0 or 1) is often desirable for certain research problems and applications, we also report in Tab. 1 the average F-measure on the hard occlusion labeling maps[5] obtained by our method with the same parameter setting for all testing data.

Finally, quantitative comparison with [44] and [35] on their datasets is shown in Tab. 2 and also demonstrates the

[4]This operation also prevents CNN from paying too much attention to the center of the label patch and assigning a high probability to it.

[5]In the experiments, the hard occlusion labeling maps is computed from the obtained probabilistic occlusion boundary map with a threshold of 0.4.

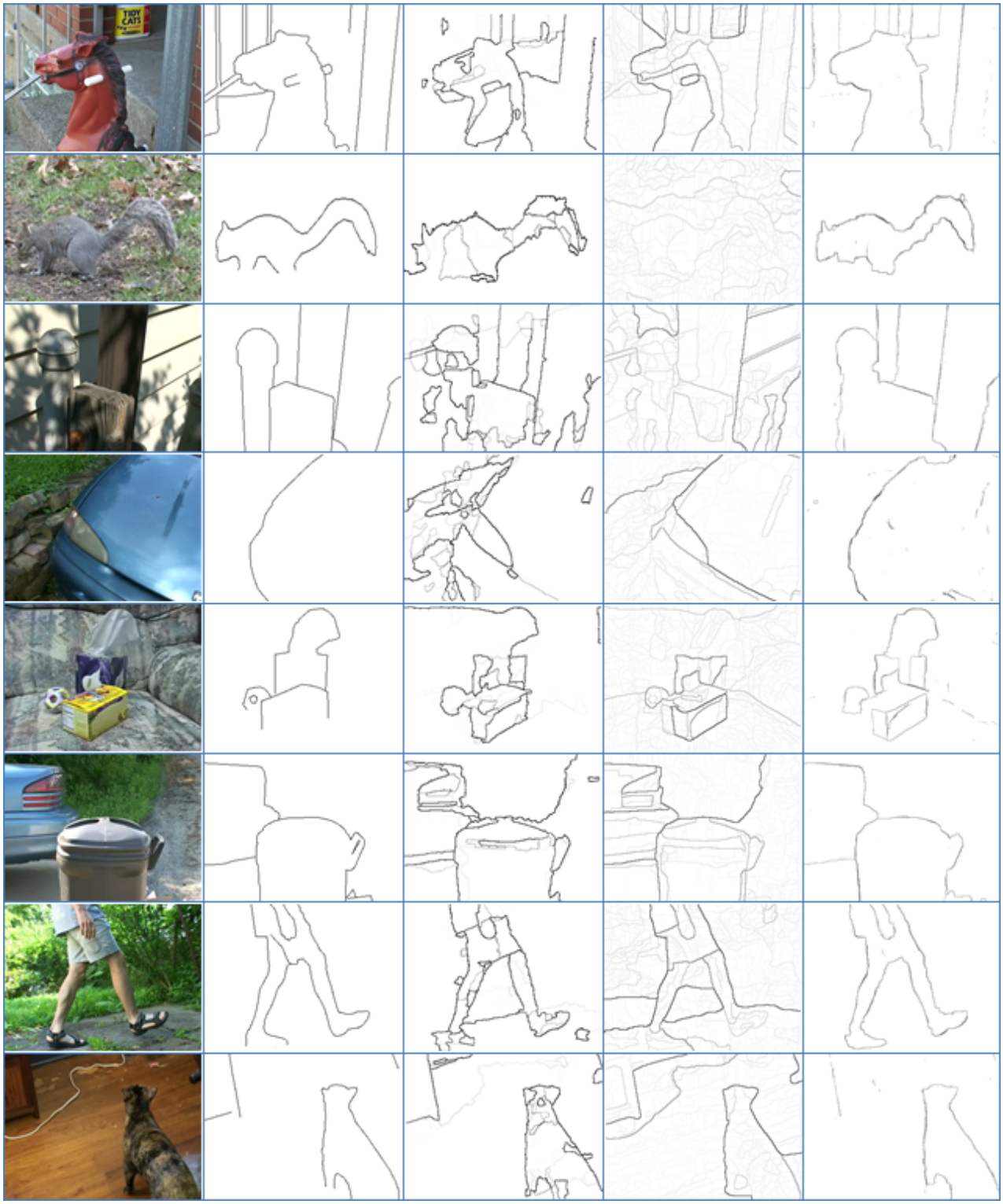

Figure 5. **Representative occlusion boundary detection results.** Each row corresponds to one testing sequence in the CMU benchmark [40] and consists of (from left to right): a reference frame, the occlusion boundary ground truth, the occlusion boundary map obtained by Stein *et al*. [40] (F-measure = 0.48), global probability boundary (gPb) map obtained by Arbelaez et al. [2] (F-measure = 0.53), and the occlusion boundary map obtained by our method (F-measure = 0.71).

Features	F-measure
Image	0.60
Image + OMF1	0.65
Image + OMF2	0.64
Image + OMF1 + OMF2	**0.71**

Table 3. **Temporal contextual exploration.** The contribution of the motion cues in the whole boundary detector is demonstrated.

superiority of our detector.

Contribution of Temporal Cues Based on *Classic-NL* [41], we performed experiments to estimate the contribution of each type of occlusion motion context to our algorithm. We can see from Tab. 3 that: (i) motion contexts are important cues that have a large impact on the performance of the occlusion boundary detection, and (ii) the two motion features used in our method both significantly contribute to the performance of the method, and jointly using them achieves the highest accuracy. Furthermore, in order to evaluate how the method's performance depends on the accuracy of optical flow computation, in Tab. 4, we report the F-measures using three typical optical flow algorithms proposed by: Broxs *et al*. [10], Sun *et al*. [41] and Weinzaepfel *et al*. [50]. These similar F-measures demonstrate that our method is quite robust with respect to the choice of optical flow algorithm.

Optical Flow	F-measure
LDOF [10]	0.68
Classic-NL [41]	0.71
Deepflow [50]	0.69

Table 4. **F-measures using different optical flow algorithms.**

Different CNN Frameworks The way contextual information is explored via local patches is a key factor of the method, since the edge and node potentials in the final CRF framework are related to the contextual information aggregated by CNNs. In order to validate our choice (*i.e*., *L2S*: we learn the mapping from a large patch "L" to a small patch "S" with the same center with "L"), we compare it with four variants, which are: (i) *L2P*: we learn the mapping from "L" to the pixel at the center of "L"; (ii) *L2SP*: we learn the mapping from "L" to "S" by independently learning the mapping to each individual pixel located within "S"; (iii) *L2L*: we learn the mapping from "L" to "L"; and (iv) *S2S*: we learn the mapping from "S" to "S". The results are shown in Tab. 5, from which we observe that: (i) the F-measure by S2S is obviously lower than that of other methods since the small input image patches contain much less contextual information and the contextual information of the surrounding area is ignored; (ii) compared to L2S, the CNN is slightly less effective at extracting discriminative spatial contextual information when learning the mapping from "L" to its center pixel, each individual pixel located within "S", and "L" itself (L2P, L2SP, and L2L). We explain these differences as follows: (i) L2P and L2SP: the CNN concentrates more on learning the differences between input samples to binary classify large patches and ignores correlations around occlusion edges within local image patches; (ii) L2L: on the one hand, the fixed training set becomes over sparse when the size of labeling map is too large, in which case the CNN can only learn superficial structural features; on the other hand, contextual correlation between the labeling of pixels and the observations from the surrounding area is not considered; and (iii) L2S: it properly handles the aforementioned issues exhibited in the variants so as to achieve better discriminative structured features.

Mapping methods	F-measure
L2P	0.66
L2SP	0.65
L2L	0.66
S2S	0.63
L2S	**0.71**

Table 5. **F-measures using different mapping methods**.

4. Conclusion

In this paper, we aim to exploit contextual information, including local structural boundary patterns, observations from surrounding regions, temporal context, and soft contextual correlations between neighboring pixels to improve performance of occlusion boundary detection. Computed occlusion motion cues and color cues are fed into a CNN framework to obtain a probabilistic occlusion boundary map on a small patch from a large patch with the same center and also to aggregate deep contextual features. Based on these, a CRF model is then adopted to achieve global occlusion boundary estimation. Our detector significantly outperforms the current state-of-the-art (*e.g*., F-measure increases from 0.62 [27,44] to 0.71 on the CMU benchmark). Last but not least, we empirically demonstrate the importance of the temporal contextual cues and the advantage of our approach to exploring contextual information.

Acknowledgment

This work is partly supported by Australian Research Council Projects (DP-140102164, FT-130101457, LE140100061) and CNRS INS2I-JCJC-INVISANA.

References

[1] N. Apostoloff and A. Fitzgibbon. Learning spatiotemporal t-junctions for occlusion detection. In *CVPR*, 2005.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, 2011.

[3] A. Ayvaci and S. Soatto. Detachable object detection: Segmentation and depth ordering from short-baseline video. *IEEE TPAMI*, 34(10):1942–1951, 2012.

[4] Y. Bengio, I. J. Goodfellow, and A. Courville. Deep learning. Book in preparation for MIT Press, 2015.

[5] I. Biederman. *On the semantics of a glance at a scene*. 1981.

[6] M. J. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In *ECCV*, 1992.

[7] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *IJCV*, 38(3):231–245, 2000.

[8] A. Blake, P. Kohli, and C. Rother. *Markov random fields for vision and image processing*. Mit Press, 2011.

[9] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010.

[10] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE TPAMI*, 33(3):500–513, 2011.

[11] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *ECCV*. 2004.

[12] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[13] X. Fan, K. Zheng, Y. Lin, and S. Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *CVPR*, 2015.

[14] D. Feldman and D. Weinshall. Motion segmentation and depth ordering using an occlusion detector. *IEEE TPAMI*, 30(7):1171–1185, 2008.

[15] F. Galasso, N. S. Nagaraja, T. Jimenez Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013.

[16] Y. Ganin and V. Lempitsky. Nˆ 4-fields: Neural network nearest neighbor fields for image transforms. In *ACCV*. 2014.

[17] J. Gibson. *The perception of surface layout: A classification*. Unpublished "Purple Perils" essay, Nov 1968.

[18] X. He and A. Yuille. Occlusion boundary detection using pseudo-depth. In *ECCV*. 2010.

[19] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[21] D. Hoiem, A. N. Stein, A. Efros, M. Hebert, et al. Recovering occlusion boundaries from a single image. In *ICCV*, 2007.

[22] A. Humayun, O. Mac Aodha, and G. J. Brostow. Learning to find occlusion regions. In *CVPR*, 2011.

[23] J. Jia, Y.-W. Tai, T.-P. Wu, and C.-K. Tang. Video repairing under variable illumination using cyclic motions. *IEEE TPAMI*, 28(5):832–839, 2006.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *ACM Multimedia*, 2014.

[25] P. Kontschieder, S. Rota Buló, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. In *ICCV*, 2011.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[27] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *ECCV*. 2012.

[28] S. Li, J. Xing, Z. Niu, S. Shan, and S. Yan. Shape driven kernel adaptation in convolutional neural network for robust facial trait recognition. In *CVPR*, 2015.

[29] T. Malisiewicz and A. Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *NIPS*, 2009.

[30] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TPAMI*, 26(5):530–549, 2004.

[31] H. Myeong, J. Y. Chang, and K. M. Lee. Learning object relationships via graph-based context model. In *CVPR*, 2012.

[32] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[33] A. Oliva and A. Torralba. The role of context in object recognition. *TICS*, 11(12):520–527, 2007.

[34] A. Owens, C. Barnes, A. Flint, H. Singh, and W. Freeman. Camouflaging an object from many viewpoints. In *CVPR*, 2014.

[35] S. H. Raza, A. Humayun, I. Essa, M. Grundmann, and D. Anderson. Finding temporally consistent occlusion boundaries in videos using geometric context. In *WACV*, 2015.

[36] M. E. Sargin, L. Bertelli, B. S. Manjunath, and K. Rose. Probabilistic occlusion boundary detection on spatio-temporal lattices. In *ICCV*, 2009.

[37] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[38] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition*. 2014.

[39] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *CVPR*, 2015.

[40] A. N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *IJCV*, 82(3):325–357, 2009.

[41] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *IJCV*, 106(2):115–137, 2014.

[42] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. *CVPR*, 2015.

[43] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[44] P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011.

[45] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, 2015.

[46] A. Torralba, K. P. Murphy, W. T. Freeman, M. Rubin, et al. Context-based vision system for place and object recognition. In *ICCV*, 2003.

[47] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE TPAMI*, 32(10):1744–1757, 2010.

[48] C. Wang, N. Komodakis, and N. Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *CVIU*, 117(11):1610–1627, 2013.

[49] X. Wang, D. F. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. 2015.

[50] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

[51] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Learning to detect motion boundaries. In *CVPR*, 2015.

[52] Y. Weiss. Comparing the mean field method and belief propagation for approximate inference in mrfs. *Advanced Mean Field Methods - Theory and Practice*, pages 229–240, 2001.

[53] T. Xiang and S. Gong. Model selection for unsupervised learning of visual context. *IJCV*, 69(2):181–201, 2006.

[54] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.