Pattern Recognition Letters 89 (2017) 73–80

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Occlusion detecting window matching scheme for optical flow estimation with discrete optimization

Kyong Joon Lee[a,*], Il Dong Yun[b]

[a] Department of Radiology, Seoul National University Bundang Hospital, Seongnam 13620, Korea
[b] Division of Computer and Electronic Systems Engineering, Hankuk Universuity of Foreign Studies, Yongin 17035, Korea

ARTICLE INFO

Article history:
Received 5 September 2016
Available online 14 February 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Optical flow
Occlusion detection
Window-matching

ABSTRACT

Occlusion detection plays an important role in optical flow estimation and vice versa. We propose a single framework to simultaneously estimate flow and detect occlusion using novel support-weight based window matching. The proposed support-weight provides an effective clue to detect occlusion based on the assumption that the occlusion occupies relatively small portion in the window. By applying a coarse-to-fine approach we successfully address non-small occlusion problems as well. The proposed method also presents reasonable estimation for the flow for the occluded pixels. The energy model with the matching cost and flow regularization cost is optimized by an efficient discrete optimization method. Experiments demonstrate our method improves estimated flow accuracy compared to the method without occlusion detection, particularly on motion boundaries. It also yields highly competitive occlusion detection results, outperforming the previous state-of-the-art methods.

1. Introduction

In estimating optical flow between a reference image and a target image, *occlusion* refers to a certain region of the reference image that does not correspond to any region in the target image due to movement of objects and/or view change. Unless properly defined, occlusion degrades the quality of estimation, particularly on object boundaries, and may lead to severe performance degeneration in many applications of optical flow estimation; for example, frame interpolation [1,2], motion segmentation [3,4], motion layer ordering [5], and motion compensated coding [6,7].

A convincing method to find exact occlusion is grasping exact motion of all objects in the images; inversely, if we obtain exact occlusion in advance, the accuracy of flow estimation will be significantly improved. In practice, neither the exact motion nor the exact occlusion is provided in advance, thus it is very challenging to obtain highly accurate optical flow and occlusion at the same time. We address this challenge with a novel window matching scheme on an unified discrete MRF (Markov Random Field) framework.

1.1. Related work

Various approaches have been presented for jointly estimating optical flow while detecting occlusion. Many of them employ individual sequential steps as following: (1) calculating optical flow as if occlusion does not exist, (2) finding occlusion based on the estimated flow, and then 3) iterating the previous two steps until convergence. One simple approach to detect occlusion given flow estimation, is thresholding the residual of subtracting warped target image from reference image [8]. In [9], authors introduce a probabilistic criterion employing histogram of image contents, and alternately calculate flow and visibility through the EM-algorithm. Alvarez et al. define occlusion by checking symmetric consistency of forward and backward flows [10]. Another method in [11] also detects occlusion by cross-checking the bi-directional flows, utilizing a coarse-to-fine approach with discrete optimization. To reduce inherent computational complexity, it estimates movement of similar pixel groups (i.e., super-pixel) with over-segmenting input images. A method in [12] utilizes observation that a certain point in a target image could probably be occluded if the point is accessible by multiple pixels in a reference image through forward warping. It finally refines the estimated flow with the probability map of accessibility. All those approaches, however, may suffer from the fact that they depend on the initial flow result which could be incorrectly estimated in the occluded area; and subsequent iterations may also yield erroneous results accordingly. Moreover, they

* Corresponding author.
E-mail addresses: kjoon31@gmail.com (K.J. Lee), yun@hufs.ac.kr (I.D. Yun).

http://dx.doi.org/10.1016/j.patrec.2017.02.009
0167-8655/

require additional computational cost for obtaining the backward flow or the occlusion probability map.

Occlusion has also been recognized as a significant issue in stereo matching problems. Zitnick et al. proposes to iteratively update a 3D disparity array using uniqueness and smoothness constraints, detecting occlusion by thresholding [13]. The uniqueness constraint implies that each pixel in a target image should have at most one correspondence in a reference image. An approach in [14] shows promising results by applying the Graph-cuts algorithm [15] to efficiently enforce the uniqueness constraint. A method in [16] uses backward disparity and visibility maps to detect symmetric occlusion using iterative optimization with the Belief Propagation [17]. These methods generally find good solutions in the discrete sample spaces; however, in the *two-dimensional* flow estimation, the size of the sample space exponentially increases, leading to very high computational complexity unless efficiently managed.

Meanwhile, Ballester et al. presented an assumption that an occluded pixel may be visible in the previous frame of a reference frame [18]. But their approach is limited to the case that multiple frames are provided and motion across the frames is relatively simple. A recent method shown in [19] utilizes over-segmentation to find image layers with respective movements, detecting occlusion with local ordering of the layers. While this method can address large occlusion issues on textureless regions, it may yield over-simplified flow estimation depending on performance of the over-segmentation. Fortun et al. also presented an approach to manage large occlusion problems [20,21]. They first compute local flow candidates on non-occluded regions, and then fill in a large occlusion based on the candidates close to the occlusion. However, filling-in may fail if no region is close enough or multiple confusing region candidates exist.

In [22], authors showed a new model incorporating a cost for occlusion which is supposed to be very sparse in the input images within infinitesimal time interval. While this method presents state-of-the-art performance in detecting occlusion, it degenerates performance of flow estimation as the process iterates. In addition, the performance can be very sensitive to a threshold value controlling sparseness of the occlusion. Another work [23] also applies the sparsity constraint estimating flow as well as occlusion; but the algorithm eventually depends on the weight map obtained from motion inconsistency, yielding insufficient performance to be state-of-the-art.

2. Proposed approach

This work aims to simultaneously estimate optical flow and detect occlusion within a single optimization framework. Our method does not iterate through flow estimation and occlusion detection. Compared to the previous state-of-the-art method [22] without the iteration, the proposed approach does not degrade the performance of optical flow estimation, indeed it does not require sensitive threshold parameter tuning.

2.1. Unified optimization framework

To this end, we propose a discrete MRF framework with novel implication of graphical nodes. In previous approaches employing the discrete framework [24–26], each pixel in a reference image is mapped to a node representing a 2D displacement vector. In contrast, a node in our framework represents a *3D* vector, comprising a 2D displacement vector and *occlusion status*. Our contribution also lies in providing an efficient method to find the optimal solution for the proposed framework.

The proposed framework defines reasonable matching cost for the new type of displacement vector with the occlusion status turned *on*. In addition to conventional intensity consistency cost on each pixel, we define a new cost for the case that the pixel is occluded but matched to a certain position in a target image. These two costs should be well balanced not to find a trivial solution (e.g., all pixels are occluded.)

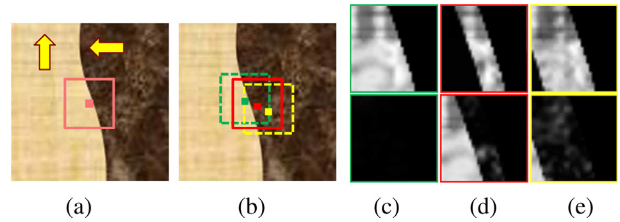

Fig. 1. Support-weight illustration for occluded pixel. A reference frame (a) and a target frame (b) contain two moving objects; the foreground object (in bright brown) moving to top, while the background object (in dark brown) moving to left. The center pixel with a support-window in the reference frame, shown in pink, is occluded in the target frame. Three matching candidate points with support-windows in the target frame are located in the foreground (green,) the occluded region (red,) and the background (yellow,) respectively. In (c), (d), and (e), support-weights for these three points are depicted for the normal weight (top row,) and for the occlusion weight (bottom row.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2. Matching cost for occluded pixel

The matching cost in this work is calculated through comparing windows based on support-weight scheme [26–28] (The *window* refers to local neighbourhoods of the pixel to be matched.) The support-weight accentuates pixels in the window in the calculation, if the pixels belong to the object that the central pixel belongs to.

When a pixel in a reference image is occluded, the pixel does not match to any position in a target, and its actual matching cost is hard to define. Avoiding this difficulty, previous approaches [8,16,22] simply assign a constant cost for occlusion candidates. However, the optimal constant cost is unknown and the resulting flow estimation for the occluded pixel mostly depends on regularization with neighbouring flows.

We assign a reasonable cost for an occluded pixel utilizing the local neighbourhoods. We observe that non-occluded neighbors in the identical object of the occluded pixel can be employed to estimate correct matching of the pixel. Based on this observation, we divide the support-weight into two different types as follows: Denoting w^{ref} and w^{tar} as the support-weights for windows in a reference image and a target image respectively, we define $w^{ref} \cdot w^{tar}$ as *normal weight* , and $w^{ref} \cdot (1 - w^{tar})$ as *normal weight*.

Fig. 1 presents an example of the support-weights (normal and occlusion weights) for an occluded pixel. The bright region represents background object while the dark region represents foreground object. The foreground in the reference (a) moves to left in the target (b). The pixel shown in pink in (a) is occluded in (b). The top and the bottom rows of (c),(d), and (e) illustrate the normal weight and the occlusion weight respectively, for the cases that the window is located in the background (c), the occlusion (d) and the foreground (e) in the target. Highly weighted part of each window is shown in high intensity, which we refer as *effective area*.

When the window is located in the occluded region as in (d), the effective area from the normal weight appears identical to the occlusion itself, assigning high weights on the pixels in the occlusion when calculating the matching cost. In contrast, the effective area from the occlusion weight exactly illustrates the non-occluded background area excluding the occlusion in the cost calculation, which can be a good clue for correct matching. In other cases when the window is located in the foreground (c) or in the back-