Occlusion Refinement for Stereo Video Using Optical Flow

Dmitry Akimov, Alexey Shestov, Alexander Voronov, Dmitriy Vatolin
Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, Russia
{dakimov, ashestov, avoronov, dmitriy}@graphics.cs.msu.ru

Abstract

In various areas of processing video in 3D format the precise occlusion mask is highly required, for example for high quality stereo-to-multiview conversion with background restoration or in quality metrics for stereo video. Stereo occlusions detected by many of existing algorithms have problems with width, alignment with objects borders and smoothness. In this paper we propose a novel technique for occlusion mask refinement using Optical Flow. Constrains on occlusion width, position and smoothness are directly incorporated in our method. Experiments and tests on real video sequences demonstrate satisfactory results. This proves the effectiveness of the proposed technique.

1. Introduction

Interest in the content in 3D formats has remarkably increased recently with more and more products and services providing the ability of watching 3D. 3D is commonly understood as a type of visual media that allows feeling depth of the scene and is often referred to the stereo video.

In many areas of 3D production and stereo video quality estimation of plausible occlusions are required. For example, in the process of depth estimation, high quality stereo-to-multiview conversion with background restoration. Also, occlusion mask can be used in high-level stereo video analysis like determining scenes with swapped right and left views, estimation of occluded area filling quality for converted video or estimation of depth quality in the object edges areas.

Precise occlusion detection is a challenging task and it has been studied for years, but still there is a lot of work to do. Many methods produce the occlusions that have the following drawbacks: their width may not correspond to the difference of the disparity leftwards and rightwards the occlusion area, their position is often not aligned to the object border and their borders are not smooth. We propose a novel technique for stereo occlusion mask refinement using disparity obtained from Optical Flow. This method addresses the issues mentioned above.

The experiments on real movies helped to define advantages and drawbacks of the proposed techniques and determine further directions of algorithm improvement.

2. Related work

Several algorithms have been proposed to perform occlusion detection. Occlusions have been a concern in the optical flow estimation problem since the first global formulation was proposed by Horn and Schunck [5]. The main idea of this and the inherited methods was in solving the nonsmooth problem with primal-dual methods decoupling the matching and regularization terms. However, none of these robust flow estimation methods focus on the detection of occlusions. Another work [7] was devoted to development of more complicated variational formulation of Optical Flow that jointly computes optical flow, implicitly detects occlusions and extrapolates optical flow in occlusion areas.

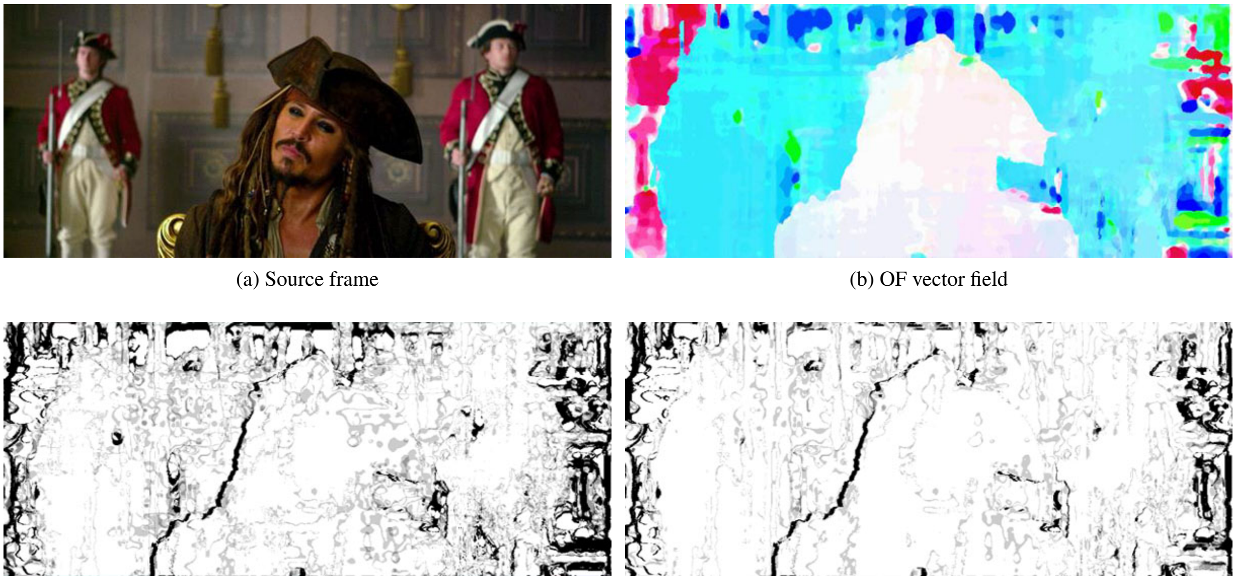
Other approaches [9, 8, 13] define occlusion detection as a classification problem and perform motion estimation in a discrete space, where it is an NP-hard problem. The result can be approximated with combinatorial optimization.

Another set of algorithms [12, 4, 6] tries to find occluded regions using training a learning based detector according to appearance, motion and depth features. The accuracy of these methods highly depends on the feature detectors precision and noise resistance.

Many authors define occlusions as the regions where forward and backward motion are inconsistent [6, 1]. Although this is problematic assumption as the motion in the occluded region is not just inconsistent, it is undefined, as there is no real "motion", in our model we will use the same assumption for the initial occlusion detection.

3. Algorithm

In this section we will describe the proposed method. The main goal of the algorithm is to estimate plausible oc-



(a) Source frame

(b) OF vector field

(c) LRC values

(d) Horizontal LRC values

Figure 1: Illustration of difference between LRC (c) and horizontal LRC (d) for source frame (a) and corresponding OF (b). The dark color corresponds to the high LRC values. The frame is taken from the movie “Pirates of the Caribbean: On Stranger Tides”.

clusions with minimum number of false positive detections, according to binocular disparity between the views. Binocular disparity (we will use the term “disparity” further) is the difference of an object location seen by the left and right eyes, resulting from the eyes’ horizontal separation. So, initial information for the algorithm comes from the disparity. To estimate it we used the Optical Flow (or OF) algorithm described in [10]. We treated disparity as the motion vector field between views. Further we will use the terms “motion vector” and “disparity” as synonyms.

First of all, given the first approximation of the occlusion area as the left-right compensation inconsistency, further clarification of the occlusion width and position can be done according to the disparity in the areas astride the supposed occlusion position.

Next, we will use the following considerations:

1. At each point of the frame outside the border areas OF vector obtain the lowest compensation error in comparison with other vectors in the point vicinity.

2. Inside the borders area foreground and background objects almost always have nonzero color difference.

3. Stereo occlusion width is equal to the difference between the disparity values of the background and foreground objects.

4. Objects boundaries are almost always piecewise-smooth, so occlusions boundaries must be piecewise-smooth too.

First and second considerations lead to the following idea: shifting the occlusion area by the expected foreground and background motion vectors, the real position of the occlusion can be determined according to the estimated compensation errors. In the case of background vector shift the area of foreground part will have a greater compensation error than background part, and vice versa for the other case.

Third consideration binds compensation errors obtained from background and foreground vectors, so they can be used not separately but together.

According to the last consideration an optimization problem under the smoothness constraint for occlusion boundaries can be posed.

This leads us to the following algorithm pipeline:

1. Disparity map and initial occlusion mask estimation .

2. Occlusion mask segmentation in order to process each occlusion area separately.

3. Background and Foreground disparity estimation.

4. Compensation error calculation.

5. Occlusion likelihood map computation (estimation of a function which describes the likelihood of placing occlusion border at current image point).

6. Occlusion border optimization problem solution.

Further, we will describe the each step of the algorithm in details.

3.1. Initial occlusion mask estimation

Left-right consistency (or LRC) is the confidence measure for inter-view Optical Flow [3]. Denoting the motion vector in the point p in the left image as $\overrightarrow{v}$ and the motion vector in the point $p + \overrightarrow{v}$ in the right image as $\overrightarrow{u}$, in ideal situation $\overrightarrow{u} + \overrightarrow{v} = \overrightarrow{0}$. So the greater $||\overrightarrow{u} + \overrightarrow{v}||$ is — the less confident is the motion vector in the point p.

As we process stereo images, in ideal situation OF vectors must be horizontal. So instead of LRC we calculate horizontal LRC, which takes into account only the x-coordinate of the vector $\overrightarrow{u} + \overrightarrow{v}$ (not its length). Using this way of LRC calculation we have less false-positives occlusions, and the number of true-negatives does not increase. A comparison between LRC and horizontal LRC is presented in Figure 1.

The next step is LRC map thresholding. This gives us a binary occlusion mask, which can de significantly improved using the morphological operations.

Median filtering is commonly used for binary mask improving. But applying the median filtering to LRC occlusion mask leads to the loss of thin occlusions. So, instead of it we perform median filtering only when it does not make less occlusion mask and then segment occlusion mask with region growing algorithm and delete small segments.

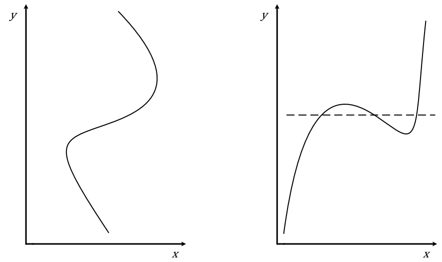
Usually there are many false positive occlusions in large smooth areas because of the poor Optical Flow quality in such regions. This type of occlusions is removed in the following way: at each point of the image the variance value of the point vicinity is calculated, then the threshold is defined as a maximum variance for smooth region.

Next, we process each row independently. Going leftwards and rightwards from the occlusion area, we check the variance value at each point of the search area or until we meet another occlusion. If the variance is less than threshold all the way, we delete current occlusion in the current row. All the parameters of this step are tuned so that they do not cause the number of true positive occlusions decrease.

The next step of morphological filtering is performing dilation and erosion in order to connect some disconnected occlusion segments. At last, we fill small holes in the occlusion mask. This gives us initial binary occlusion mask.

3.2. Occlusion segmentation

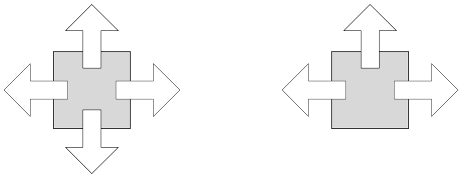
At this step we must segment occlusion mask into separate regions with vertically monotonic borders in order to correctly perform further steps of the algorithm. We call a curve vertically monotonic if for each y_0 there is only one point on the curve which y-coordinate is equal to y_0 (see Figure 2).



(a) Vertically monotonic curve

(b) Vertically non-monotonic curve

Figure 2: Example of vertically monotonic (a) and non-monotonic (b) curves.



(a) Common region growing process

(b) Upward region growing process

Figure 3: Allowed directions for segmentation propagation in common (a) and upward (b) region growing processes.

The key procedure is region growing algorithm with modified region growing process — usually regions grow in 4 directions, in our implementation regions grow in 3 directions (see Figure 3). Let us call algorithm upwards (downwards) region growing if we do not consider pixel downwards (upwards) of the current pixel . When performing upwards region growing you must process pixels from bottom to top and from top to bottom for downwards region growing.

First, we perform upwards region growing, then downwards region growing. So, we have segments, which borders are monotonic in vertical direction, but they can still have big holes. We deal with holes in the following way: going from the top row to the bottom row, through the each row, if we meet some segment for the second time, we apply downwards region growing to the first met piece of the current segment with new segment number equal to negative current segment number, then apply downwards region growing to the second met piece with another segment number. In the further cases (if we meet a piece of segment more than 2 times in one row), we do not apply downwards region growing to the firstly met piece.

In the next steps each occlusion is processed as independent object, row by row. On each step the row may be discarded from a further consideration. We will call such row as a “bad row”.

3.3. Left and right difference maps

The next step is estimation of the differences between pixels in left and right views in occlusion areas after the area shifts by the expected foreground and background motion vectors.

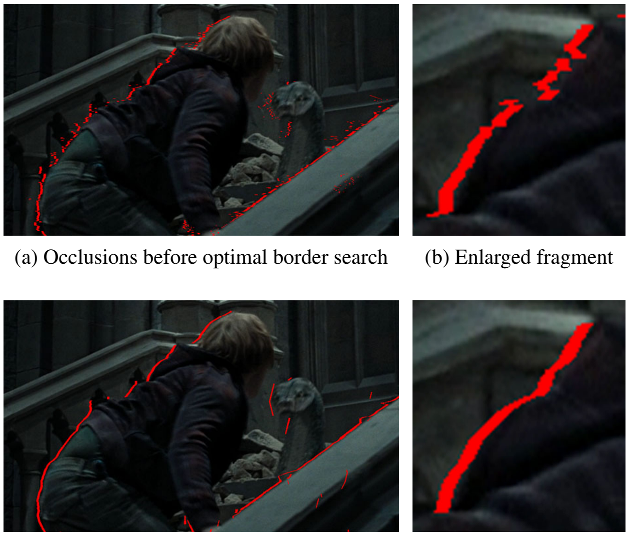
It is a common case that Optical Flow have significant errors at the object border areas. To achieve the resistance to this type of noise we suggest widening the initial occlusions leftwards and rightwards in the areas with the similar OF values around and inside the occlusion. With high probability the widened occlusion area includes the parts of the foreground and the background. That is why the OF values astride the widened occlusions more likely characterize the behaviour of the foreground and background object then the ones around the initial area.

Going leftwards or rightwards the occlusion left or right border respectively, algorithm stops if there is another occlusion or a long region of another flow (OF values in the considering region are significantly different from the values in the area near the occlusion border). If the found region width is less then predefined threshold, we treat current row as a bad row. Then we unite found regions and LRC occlusion and deal only with good rows of it. The obtained mask is called “difference mask”. This is the region for the true occlusion position search.

Next, for each row of the occlusion the approximations of the foreground and background motion are estimated as the median value of OF vectors leftwards (OF_L) and rightwards (OF_R) the occlusion border. The median is calculated using the points from the current row and neighbour rows upwards and downwards, if they are not bad. Each vector can be characterized by its frame compensation quality [11]. If there are too few vectors with the adequate quality (greater than the predefined threshold) in current calculations, we treat current row as bad.

OF_L and OF_R are used to define occlusion width $W(y)$ at each row. The width is calculated as the difference of x-coordinates of OF_L and OF_R. Absolute value of x-coordinate of OF_L must be greater than absolute value of x-coordinate of OF_R. If it is not so, we treat current row as a bad row. The other usage of the OF_L and OF_R.

Next, difference maps $Diff_L$ and $Diff_R$ are estimated using the OF_L and OF_R shifts (see Figure 4 (a) and (b)).



(a) Occlusions before optimal border search

(b) Enlarged fragment

(c) Occlusions after optimal border search

(d) Enlarged fragment

Figure 5: Example of occlusion improvement after optimal border search. The frame is taken from the movie “Harry Potter and the Deathly Hallows: Part 2”.

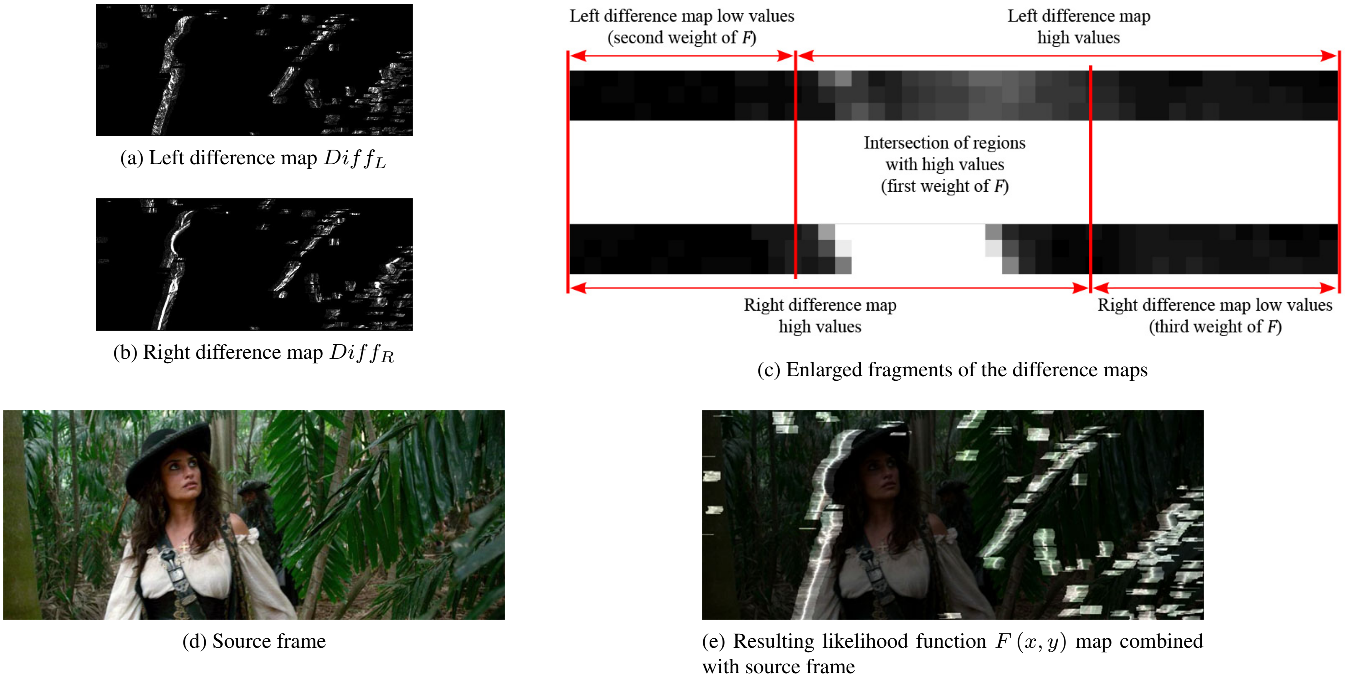
3.4. Occlusion likelihood map

Using the difference maps estimated at the previous step the likelihood value of placing the occlusion border at the point (x, y) of the difference mask is calculated as follows:

$$F(x,y) = \omega \int\limits_{x}^{x+W(y)} \left(Diff_L(\tau,y)\,\mathrm{d}\tau + Diff_R(\tau,y)\,\mathrm{d}\tau\right) - \int\limits_{border_L}^{x} Diff_L(\tau,y)\,\mathrm{d}\tau - \int\limits_{x+W(y)}^{border_R} Diff_R(\tau,y)\,\mathrm{d}\tau \qquad (1)$$

The first component reflects the consideration that occlusion area is an intersection of the $Diff_L$ and $Diff_R$ regions with high difference values, so sum of differences must be big in such regions. Sum of the last two components reflects the consideration, that $Diff_L$ values leftwards and $Diff_R$ values rightwards the occlusion must be small. The example is presented in Figure 4.

The the first approxiamtion of the occlusion border position x_{border} at each y row of the occlusion can be defined as $x_{border} = \arg\max_{x} F(x,y)$.



(a) Left difference map $Diff_L$

(b) Right difference map $Diff_R$

(c) Enlarged fragments of the difference maps

(d) Source frame

(e) Resulting likelihood function $F(x, y)$ map combined with source frame

Figure 4: Occlusion likelihood map estimation. Given the source frame (d) and corresponding difference maps $Diff_L$ (a) and $Diff_R$ (b) we estimate $F(x, y)$ (c). The frame is taken from the movie "Pirates of the Caribbean: On Stranger Tides".

3.5. Optimal occlusion border search

In this section we formulate an occlusion border optimization problem in a discrete space. The main goal is estimation of the smooth occlusion borders.

Denote $F_{norm}(x, y) = F(x_{border}, y) - F(x, y)$. For each row y we take into consideration points with $F_{norm}(x, y)$ less than predefined threshold. If there are no points with $F_{norm}(x, y)$ greater than the minimum allowed value, we treat current row as bad. If the total number of bad rows is greater than predefined threshold, the current occlusion is defined as a false positive detection and excluded from the process.

We treat each point as a node of the graph. Each node is connected to and only to the nodes of the downwards row. So we need to find the shortest path from the top row to the bottom row. The shortest path in terms of $F_{norm}(x, y)$ and spatial distance between the nodes can be found using dynamic programming methods for distance optimization problem for oriented graphs without cycles [2]. Determine the distance between the graph nodes as follows:

$$D(n_i, n_{i+1}) = (n_i - n_{i+1})^2 \cdot \left(F_{norm}(n_i)^{0.4} + F_{norm}(n_{i+1})^{0.4}\right) \cdot angle\left(\overrightarrow{n_i n_{i+1}}, \overrightarrow{n_{i-1} n_i}\right) \quad (2)$$

The bad row position is defined as a point in the straight line between border positions in the nearest good rows.

An example of optimal borders search results is presented in Figure 5.

4. Experiments

To test the proposed technique and analyse its advantages and drawbacks we used video sequences from the real movies which were initially shot in 3D format.

The obtained occlusions have several advantages comparing to standard LRC occlusions (see Figure 7):

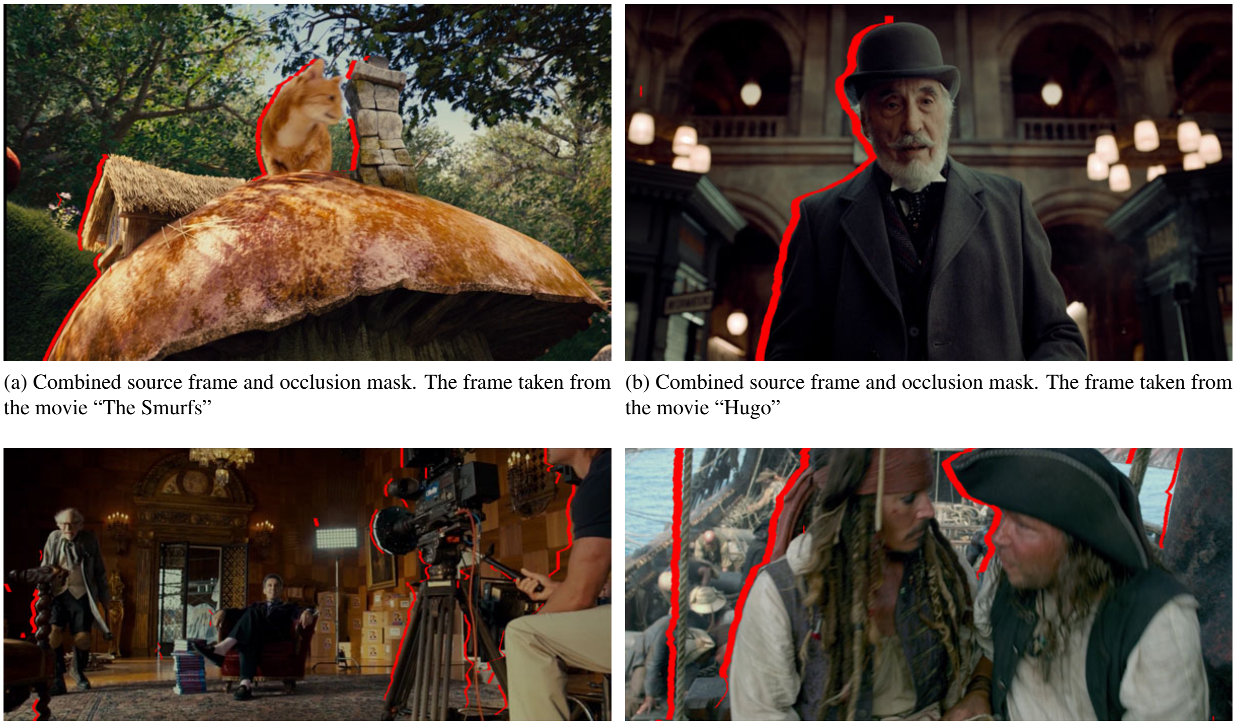
1. There are less false-positive occlusions.

2. The number of false-negative occlusions is not increased.

3. Occlusion width corresponds to the difference of the OF vectors at occlusion borders.

4. Occlusion borders follow the object borders more precisely.

5. Positions of occlusions are estimated with the quarter-pixel precision.



(a) Combined source frame and occlusion mask. The frame taken from the movie “The Smurfs”

(b) Combined source frame and occlusion mask. The frame taken from the movie “Hugo”

(c) Combined source frame and occlusion mask. The frame taken from the movie “Pirates of the Caribbean: On Stranger Tides”

(d) Combined source frame and occlusion mask. The frame taken from the movie “Transformers: Dark of the Moon”

Figure 6: Examples of algorithm results on real movies shot in 3D

4.1. Performance and speed

Algorithm was tested on Intel Corei7-2630QM CPU @ 2.00GHz, 4 cores, 8 GB RAM. The test set contained three short video sequences from five movies shot in 3D: “Smurfs”, “Hugo”, “Transformers: Dark of the Moon”, “Pirates of the Caribbean: On Stranger Tides”, “Harry Potter and the Deathly Hallows: Part 2”.

An average processing time for one frame in SD (720×480) resolution is 0.69 sec. per frame. Result for HD (1280×720) — 2.31 sec. per frame

4.2. Further improvements

The first possible direction is connection of closely placed ends of occlusions. In some cases LRC values can be not stable along the occlusion and can fluctuate around the threshold value. So some parts of occlusions may be lost and there is a reason to perform dilation of an occlusion mask.

If we dilate the whole occlusion mask we will also connect some occlusions which are indeed disconnected. So we need to connect only ends of occlusion segments.

The next important task is improvement of the occlusion and object border alignment. We can use information about orientation of the view and align right border (if we process left view) or left border (if we process right view) of occlusion with borders in image. Occlusions are not aligned with objects borders if some pieces of background and object have similar color or if background is very smooth.

Sometimes the real occlusion width is not equal the difference of OF vectors on its borders. Such cases can be detected by low values of $F(x, y)$.

Next, occlusions of different width must be processed independently, because they can be connected but belong to different objects, so joint aligning of their borders can produce wrong results. The same problem refers to the confidence.

In the current implementation of the algorithm temporal information is not taken into account, so occlusions can be temporally unstable in regions, where we can not uniquely place occlusion (occlusion rows, which have no strong maximum of $F(x, y)$).

The last area of further research is the alternative ways for initial occlusion detection, because current LRC technique produces significant number of false negative areas in occlusion map. For example, we can analyse OF in the whole image of reduced resolution and find the areas, which are candidates for occlusions.

5. Conclusion

An algorithm for occlusion estimation using OF information is proposed. Detected occlusions can be used in applications for depth estimation, high quality stereo-to-multiview conversion with background restoration or in quality metrics for stereo video: determining scenes with swapped right and left views, estimation of occluded area filling quality for converted video, estimation of depth quality on objects edges. The algorithm was tested on sequences of real movies initially shot in 3D and showed satisfactory results. The complexity is estimated. All the drawbacks and advantages are analysed and possible ways for the algorithm improvement are proposed.

6. Acknowledgements

This work is partially supported by the Intel/Cisco Video-Aware Wireless Network (VAWN) Program and by grant 10-01-00697a from the Russian Foundation of Basic Research.

References

[1] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez. Symmetrical dense optical flow estimation with occlusions detection. *Int. J. Comput. Vision*, 75(3):371–385, Dec. 2007. 1

[2] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction To Algorithms*. MIT Press, 2001. 5

[3] G. Egnal and R. P. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1127–1133, Aug. 2002. 3

[4] X. He and A. Yuille. Occlusion boundary detection using pseudo-depth. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 539–552, Berlin, Heidelberg, 2010. Springer-Verlag. 1

[5] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981. 1

[6] A. Humayun, O. M. Aodha, and G. J. Brostow. Learning to find occlusion regions. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 2161–2168, Washington, DC, USA, 2011. IEEE Computer Society. 1

[7] S. Ince and J. Konrad. Occlusion-aware optical flow estimation. *IEEE Transactions on Image Processing*, 17(8):1443–1451, 2008. 1

[8] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. *Proceedings Eighth IEEE International Conference on Computer Vision ICCV 2001*, 2(1):508–515, 2001. 1

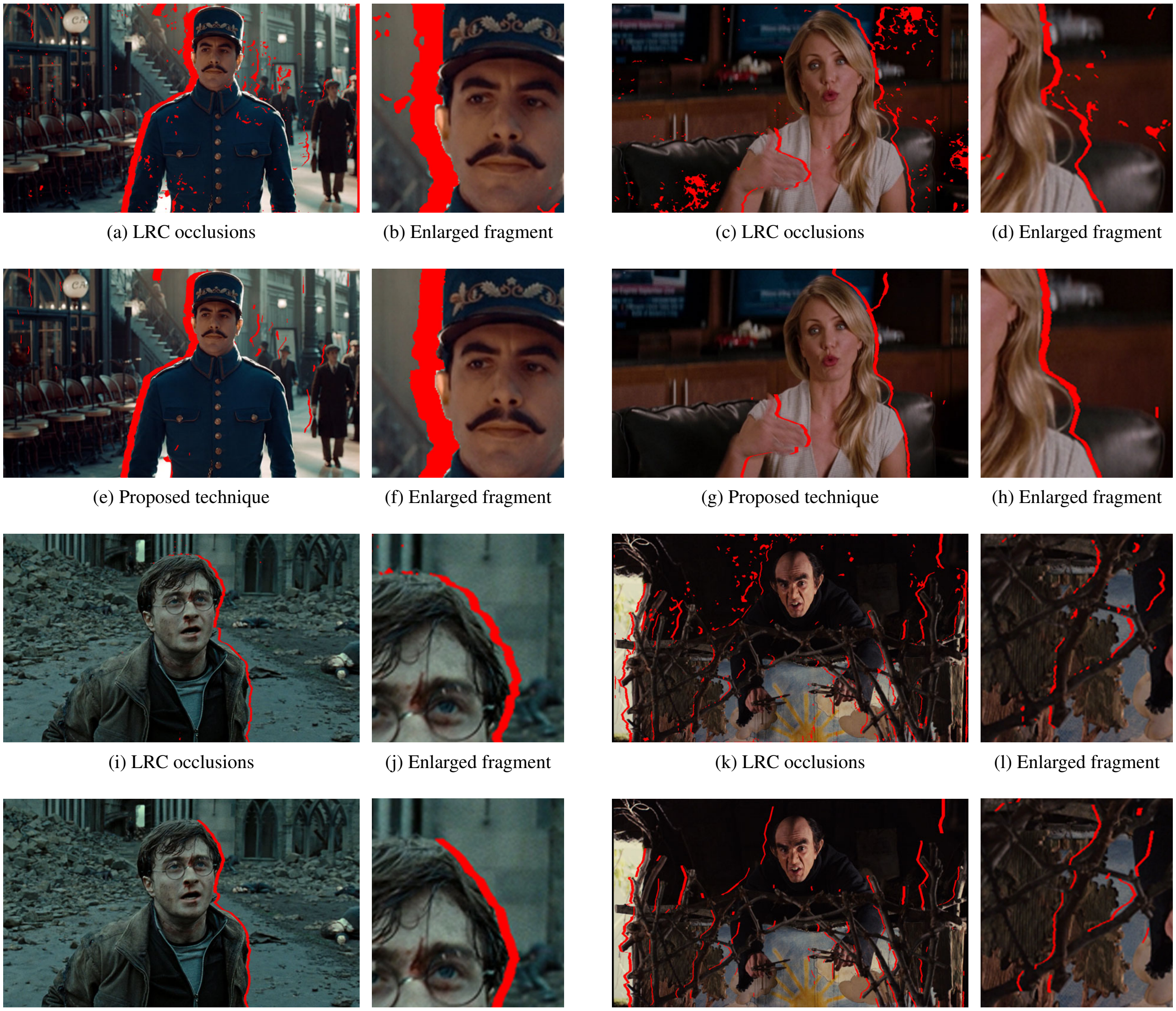
[9] K.-P. Lim, A. Das, and M.-N. Chong. Estimation of occlusion and dense motion fields in a bidirectional bayesian framework. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):712–718, 2002. 1

[10] A. S. Ogale and Y. Aloimonos. Shape and the stereo correspondence problem. *Int. J. Comput. Vision*, 65(3):147–162, Dec. 2005. 2

[11] K. Simonyan, S. Grishin, and D. Vatolin. Confidence measure for block-based motion vector field. In *Proceedings of Graphicon'2008*, pages 110–113, Moscow, Russia, 2008. 4

[12] A. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *International Journal on Computer Vision*, 82(2):325–357, April 2009. 1

[13] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '05, pages 399–406, Washington, DC, USA, 2005. IEEE Computer Society. 1



(a) LRC occlusions (b) Enlarged fragment (c) LRC occlusions (d) Enlarged fragment

(e) Proposed technique (f) Enlarged fragment (g) Proposed technique (h) Enlarged fragment

(i) LRC occlusions (j) Enlarged fragment (k) LRC occlusions (l) Enlarged fragment

(m) Proposed technique (n) Enlarged fragment (o) Proposed technique (p) Enlarged fragment

Figure 7: Comparison of the proposed technique (e–h, m–p) and the initial LRC metric (a–d, i–l). The frames are taken from the movies “Hugo” (a–b, e–f), “The Green Hornet” (c–d, g–h), “Harry Potter and the Deathly Hallows: Part 2” (i–j, m–n), “The Smurfs” (k–l, o–p).