Comparison of stereo inspired optical flow estimation techniques

Michaela Musterfrau
Vienna University of Technology
Michaela.Musterfrau@tuwien.ac.at.at

ABSTRACT

The similarity of the correspondence problems in optical flow estimation and disparity estimation techniques enables methods to adopt knowledge from the Stereo Vision literature to enhance optical flow estimation. This knowledge can be used in the three key-problems of optical flow estimation: the motion representation, the estimation criteria and the optimization. We describe and compare two existing methods, which borrow from the Stereo Vision literature to respectively address one of these key problems. The first method uses a discrete optimization algorithm, which is also applied by top-performing stereo approaches, to fuse candidate solutions. The second method includes color-segmentation or more precisely a segment-wise representation into the estimation process, what has proven to be useful to stereo approaches. In this paper we examine the respective energy functions, motion models and optimization methods. We validate the performance of the described methods on various benchmark datasets which are offered by the Middlebury optical flow website. In this context, we show that the described methods go beyond traditional techniques and are able to cope with common problems in optical flow estimation, like textureless regions, occlusions and the preservation of motion discontinuities. Finally, we conclude and highlight strengths and weaknesses of both techniques.

KEYWORDS
optical flow, motion, stereo vision, optimization, computational perception, evaluation

1 Introduction

Dense and accurate optical flow fields have various applications in the field of visual computing, such as robot vision, compression and extracting motion information [3]. Traditional approaches like the Horn and Schunk (HS) [4] or the Lucas and Kanade (LK) [5] method fail at flow borders or in textureless regions and are therefore not accurate enough for the above mentioned domains.

To overcome these problems, methods, such as [1] or [2], borrow from the Stereo Vision literature, which in fact is concerned with a closely related correspondence problem. Moreover, in the case of disparity estimation a corresponding pixel in one image lies along a single line, the epipolar line, of the other one. Optical flow estimation, on the other hand, is more general. In the scope of two adjacent frames a pixel can move with arbitrary velocity in any direction. Hence optical flow estimation is concerned with the entire image. However, in practice assumptions are formulated to reduce the possible solutions.

In this paper two methods which are inspired by stereo vision techniques are discussed and compared. The first one, *Fusion Flow* [1], subsequently fuses candidate solutions using discrete optimization, which is used by top-performing stereo techniques.

The second method, *Segmentation Flow* [2], relies on a variation model based on color-segmentation, which has proven to be useful to stereo approaches as well. It merges flow data from segments with pixel-wise computed flow fields.

We take a closer look at the two techniques mentioned above, their respective energy functions and motion models. Additionally the different ways to obtain and combine the initial flow data and how they influence the final result are examined. Finally the methods are compared in terms of their respective performance on various benchmark datasets, which are offered by the Middlebury optical flow website [3]. The latter validates whether the methods cope with common problems in optical flow estimation, such as textureless regions, occlusions, non-rigid motion, over-smoothing and the preservation of motion discontinuities.

The rest of this paper is organized as follows. Section 2 describes the two mentioned stereo inspired optical flow estimation techniques in detail. In Section 3 the results of these methods are presented and compared. Finally, Section 4 draws conclusions.

2 Stereo inspired optical flow

This section is concerned with two optical flow estimation techniques which borrow from the Stereo Vision literature in different ways. The first method does that by formulating a non-convex energy function and optimizes it discretely. The second method relies on segmentation based stereo approaches which promise useful results in textureless and occluded regions.

2.1 Fusion Flow[1]

The basic idea behind *Fusion Flow* is to avoid purely continuos optimization algorithms, which are likely to get trapped in a poor local minimum, and at the same time formulate a non-convex energy function [1]. Still, as a first step, initial solutions are created by applying the HS [4] and the LK [5] method in a coarse-to-fine manner and varying their parameters and number of levels of detail. Together with shifted copies of these solutions and 64 constant flow fields, subsequent steps of the algorithm are based on about 200 initial flow fields which complement each other.

The problem of obtaining an optimal optical flow field $\mathbf{f}$ based on these initial solutions is expressed by the following energy function:

$$E(\mathbf{f}) = \sum_{\mathbf{p}\in\Omega} D_{\mathbf{p}}(f_{\mathbf{p}}; I^0, I^1) + \sum_{(\mathbf{p},\mathbf{q})\in N} S_{\mathbf{p},\mathbf{q}}(f_{\mathbf{p}}, f_{\mathbf{q}}) \quad (1)$$

Here I^0 and I^1 are two adjacent images, $f_{\mathbf{p}} = (u_{\mathbf{p}}, v_{\mathbf{p}})$ is a flow vector at pixel $\mathbf{p}$, N is the set of all pixel-pairs and Ω denotes the domain of I^0. The data term $D_{\mathbf{p}}(.)$ applies the non-linearized color constancy assumption on higher spatial frequencies of the images (H, difference of original image and its Gaussian filtered version) and penalizes large color changes using the Geman-McClure penalty function p(.):

$$D_{\mathbf{p}}(f_{\mathbf{p}}; I^0, I^1) = p(||H^1(\mathbf{p} + f_{\mathbf{p}}) - H^0(\mathbf{p})||) \quad (2)$$

The spatial term $S_{\mathbf{p},\mathbf{q}}(.)$ applies the smoothness assumption using pairwise Markov random fields:

$$S_{\mathbf{p},\mathbf{q}}(f_{\mathbf{p}}, f_{\mathbf{q}}) = p_{\mathbf{p},\mathbf{q}}(\frac{u_{\mathbf{p}} - u_{\mathbf{q}}}{||\mathbf{p} - \mathbf{q}||}) + p_{\mathbf{p},\mathbf{q}}(\frac{v_{\mathbf{p}} - v_{\mathbf{q}}}{||\mathbf{p} - \mathbf{q}||}) \quad (3)$$

The fractions result in low values where nearby pixels $\mathbf{p}$ and $\mathbf{q}$ have similar flow vectors $(u_{\mathbf{p}}, v_{\mathbf{p}})$ and $(u_{\mathbf{q}}, v_{\mathbf{q}})$. Function $p_{\mathbf{p},\mathbf{q}}(.)$ penalizes differences according the weighted, negative log of the Student t-distribution. Assuming that flow changes appear at the same spatial location as color changes the weights lower the result for similar color values at $I^0(\mathbf{p})$ and $I^0(\mathbf{q})$.

The final non-convex energy function is minimized with a graph cut method based on fusion moves, for details see [1]. The first iteration subsequently merges the initial HS and LK solutions with the current flow field. Before the second iteration starts, constant flow fields are added. They are derived from clusters, which are generated applying the k-means algorithm on the current solution.

Once the discrete optimization has found an optimal solution, it is improved by an additional continuos optimization step. Finally, areas without variation in the initial solutions are enhanced with a local gradient decent.

2.2 Segmentation Flow[2]

Segmentation Flow proposes a segmentation based approach and is thereby fundamentally different than the above described method. In the case of disparity estimation, including segmentation in the estimation process promises to produce reasonable solutions in textureless and occluded regions. The adaption for optical flow, however, involves further issues, like defining motion models for segments and handling non-rigid motion.

Segmentation Flow addresses these problems in three steps. First a variational optical flow computation obtains an initial flow field, see [2] for further information. The second step starts with a Mean-shift segmentation of both the input images with respect to color and the optical flow field from the previous step. Then color segments are further split according to the flow segments. The parametric motion in final segments relies on an affine motion model and is estimated in

[1] This section summarizes the optical flow estimation technique presented in [1].

[2] This section summarizes the optical flow estimation technique presented in [2].

a coarse-to-fine manner. The motion estimation is again expressed with an energy function (see [2] for detail), which in this case is minimized by the Quasi-Newton method. Applying the optical flow computations as well on the revised image sequence yields two pixel-wise and two segment-wise flow fields.

In order to combine these fields, the last step introduces a confidence map, which aims to detect corrupt estimates from the second step. The map consists of three parts, which are described in detail in [2] and are briefly outlined below.

The occlusion value $O(\mathbf{p})$ for a pixel $\mathbf{p}$ is set to 1 (occluded) or to 0 (otherwise). The color of an occluded pixel in the previous image $I^0(\mathbf{p})$ does not match the color of its corresponding pixel, according to the initial optical flow $f^0_{\mathbf{p}}$, in its successor $I^1(\mathbf{p} - f_{\mathbf{p}})$.

A pixel-wise term $C_p(\mathbf{p}, f^s_{\mathbf{p}})$ either yields a constant penalty value for an occluded pixel or a value representing the motion error $E_p(\mathbf{p}, f^s_{\mathbf{p}})$. The latter is the product of a color constancy term, which is defined over the segment-wise optical flow field $f^s_{\mathbf{p}}$ $(s > 0)$ and a left-right check between the segment-wise flow of the original and the revised sequence. Note that the term is still pixel-wise, since flow values of individual pixels are interpolated from the segment-wise flow.

A segment-wise term $C_s(s, f^s_{\mathbf{p}})$ compares the segment-wise and the pixel-wise flow values of visible pixels, which belong to a segment s. This difference is weighted by the confidence value for the initial flow $E_p(\mathbf{p}, f^0_{\mathbf{p}})$.

In the final confidence map $\mathbf{conf}(\mathbf{p})$ low confidence at a pixel is a combination of reliable initial flow values, differing flow in its associated segment $s(\mathbf{p})$ and the pixel-wise motion confidence.

$$\mathbf{conf}(\mathbf{p}) = C_s(s(\mathbf{p}), f^s_{\mathbf{p}})C_p(\mathbf{p}, f^s_{\mathbf{p}}) \tag{4}$$

Once the confidence map is constructed, the final energy function can be expressed as

$$\begin{aligned} E(\mathbf{f}) = \int_\Omega &(1 - O(\mathbf{p}))\Psi(\mid I^1(\mathbf{p} + \mathbf{f}) - I^0(\mathbf{p}) \mid^2) \\ &+ \beta \mathbf{conf}(\mathbf{p}) \parallel \mathbf{f} - f^s_{\mathbf{p}} \parallel^2 \\ &+ \alpha\Psi(\parallel \nabla u_{\mathbf{p}} \parallel^2 + \parallel \nabla v_{\mathbf{p}} \parallel^2) d\mathbf{p}, \end{aligned} \tag{5}$$

where $\Psi(.)$ is the Total Variation regularizer, ∇ is the first-order derivate operator and $\mathbf{f}$ denotes the final optical flow field. Examining Equation 5 in detail, the first term applies the color constancy assumption to not occluded pixels and the last term the smoothnes assumption. The summand in the middle controls the influence of the segmented-flow on the final result, i.e. for large confidences the final flow resembles the segment-wise flow. The parameters α and β are used to further adjust the terms.

The energy function $E(\mathbf{f})$ is minimized by solving the corresponding Euler-Lagrange equations, which results in the final optical flow field.

3 Results and Comparison

The two previously described methods have the common goal to obtain optical flow fields, which overcome problems of traditional flow estimation. Namely, these are the aperture problem, textureless regions, camera noise, motion discontinuities, occlusions, large motion and illumination changes. To validate whether the described methods cope with these problems we compare their respective performance on the Middlebury optical flow dataset [3]. According to the different error metrics offered by [3] the comparison can be carried out on flow errors and interpolation errors. Additionally, we address the runtimes.

For each of these categories this section gives the results of the presented techniques in table form. Then we discuss the given measures and highlight the method's strengths and weaknesses.

3.1 Flow Errors

When comparing the performance of optical flow algorithms, the result's deviation of the ground-truth has to be considered. In [3] this issue is addressed by two errors, the average Angle Error (AE) and the average Endpoint Error (EE). The results of the techniques are given in Table 1.

In terms of both errors *Fusion Flow* is in summary more promising than *Segmentation Flow*. When we examine the respective EE measures in detail (see Table 1) the latter has only a lower error value for discontinuities of the sequence **Wooden**. Since this sequence contains rigidly moving objects and little texture the positive impact of the the rigid segment-wise flow field on motion discontinuities is reflected in this result. This statement is supported by the AE at largely rigid scenes (**Grove**, **Yosemitte**, **Teddy**) and the EE of **Yosemitte** and **Teddy**. However, in six out of eight sequences the average EEs of *Fusion Flow* are between four and 15 pixels lower than respective values of the other technique. Since the EE does not downweight large motion computing the error [3], the comparison in terms of EE is more convincing.

When we examine the measures over the eight Middlebury datasets [3], *Fusion Flow* has the highest EEs on **Urban**, **Grove** (see Table 1) and **Teddy** (all: 1.07, dis: 2.07, un: 1.39), which have strong discontinuities (< 20 pixel) and large

Table 1: This extract from the Middlebury dataset measures [3] shows the EE of *Fusion Flow* (F) and *Segmentation Flow* (S) for three real-world (**Army**, **Schefflera**, **Wooden**) and two synthetic (**Grove**, **Urban**) scenes. The error is computed over three different error masks: everywhere (all), at motion discontinuities (dis), in untextured regions (un). To enable the comparison with traditional motion estimation techniques, we exemplarily list the error values for the HS method.

EE	**Army**			**Schefflera**			**Wooden**			**Grove**			**Urban**		
	all	dis	un	all	dis	un	all	dis	un	all	dis	un	all	dis	un
F	0.11	0.34	0.10	0.29	0.66	0.23	0.20	1.19	0.14	1.07	1.42	1.22	1.35	1.49	0.86
S	0.15	0.36	**0.10**	0.68	1.24	0.64	0.32	**0.86**	0.26	1.18	1.50	1.47	1.63	2.09	0.96
HS	0.22	0.55	0.22	1.01	1.73	0.80	0.78	2.02	0.77	1.26	1.58	1.55	**1.43**	2.59	1.00

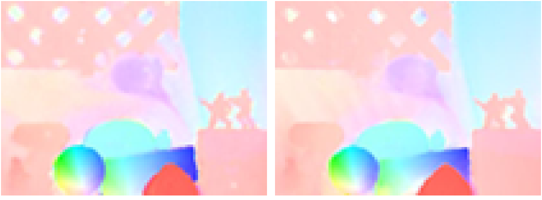

Figure 1: The images above, taken from [3], show the optical flow fields for **Army** obtained by *Segmentation Flow* (right) and *Fusion Flow* (left).

Table 2: This table (from [3]) lists the robustness (RX %) and accuracy (AX pixel) statistics based on the average EE of *Fusion Flow* (F) and *Segmentation Flow* (S). RX denotes the percentage of pixels with EEs above X. The accuracy is addressed by the X^{th} percentile for EEs up to AX.

	R0.5	**R1.0**	**R2.0**	**A50**	**A75**	**A95**
F	9.1	8.7	8.2	9.9	8.8	11.0
S	14.8	13.7	9.5	12.8	13.1	10.0

motions (<35 pixel). Furthermore, the errors indicate better performance on real-world than on synthetic scenes, which is according to [1] a result of the chosen parameters (16 for $p(.)$, $p_{\mathbf{p},\mathbf{p}}(.)$ is weighted by 0.024 for absolute differences less than 30 and 0.008 otherwise). For instance, increasing the smoothness weight by the factor 16 reduces the EE on **Yosemite** by 2.22 degrees [1]. *Fusion Flow* has the lowest values in the **Army** sequence (see Table 1), which contains only motion up to four pixels. Comparing the overall performance of this algorithm to other submitted (see [3]) estimation techniques, it is for **Mequon** ranked on place two (for un and dis rank 5), which contains non-rigid motion and texturesless regions. It also performs well on **Schefflera** (rank 7 for all, 8 for dis, 16 for un), what indicates that it can cope with little contrast, especially between foreground and background.

Segmentation Flow performs poorly on sequences with motion discontinuities, such as **Grove** and **Urban**, but has low error rates on **Army** (see Table 3.1) and **Yosemite** (all: 0.08, dis: 0.13, un: 0.12), which has few motion boundaries. In fact, this approach is top-ranked (rank 2 for all, 6 for dis, 3 for un) for the latter sequence [3]. Following this observations further and evaluating the error metrics as well as the flow fields, we conclude that this algorithm has the tendency to oversmooth motion boundaries and fine structures, which are both present in **Grove**. Furthermore, Figure 1 shows exemplarily that the flow field of *Segmentation Flow* (right) is smoother than the result of *Fusion Flow* (left) for the same sequence.

Continuing this comparison in terms of statistics, we analyze the accuracy and the robustness measures, which are offered by the Middlebury benchmark [3]. Table 2 shows that *Fusion Flow*, averaged over all test sequences, is more robust than *Segmentation Flow*. The measures for an EE of 2.0 differ about one percent, but the values drift apart with growing precision. On the other hand, the improvement from R0.5 to R2.0 indicates that *Segmentation Flow* is able to avoid gross outliners [3]. This is the result of its occlusion term and the influence of the segmentwise flow, which smooths outliers. On the contrary, its increasing accuracy from A95 to A50 suggests that this algorithm is not able to provide high accuracy. More precisely, in average over all datasets and region masks, already 50 percent of the flow vectors differ up to 12.8 pixel from the ground truth. In contrast, the measure of *Fusion Flow* for A50 is 9.9 and increases only by 1.1 pixel to A95. Additionally, it is more stable and hence more accurate than *Segmentation Flow*.

3.2 Interpolation Errors

For applications, such as novel view generation and compression, the comparison in terms of interpolation errors has to be considered. Instead of the difference between the flow field provided

Table 3: This extract from the Middlebury dataset measures [3] shows the IE of *Fusion Flow* (F), *Segmentation Flow* (S) and the HS method for regularly recorded (**Schefflera**) high-speed camera (**Basketball**, **Dumptruck**, **Evergreen**) and one synthetic (**Urban**) scenes. The error is computed over three different error masks: everywhere (all), at motion discontinuities (dis), in untextured regions (un).

IE	**Schefflera**			**Urban**			**Basketball**			**Dumptruck**			**Evergreen**		
	all	dis	un	all	dis	un	all	dis	un	all	dis	un	all	dis	un
F	3.75	5.47	1.42	4.08	5.55	3.08	6.99	13.7	2.6	8.4	19.4	1.65	8.5	13.3	1.8
HS	4.91	6.65	1.92	6.13	6.85	3.53	**6.16**	**11.9**	**2.32**	8.63	19.5	1.84	**7.91**	**12.3**	**1.73**
S	**4.17**	**6.1**	**1.59**	8.69	7.75	5.15	**6.79**	**13.2**	**2.5**	10.1	23.5	2.55	8.80	13.8	**1.72**

by an algorithm and the ground truth, we are in this case interested in the intermediate frames which are predicted using the computed flow fields. In [3] two error measures, the average Interpolation Error (IE) and the average Normalized Interpolation Error (NE), address this issue. Both measures describe the difference between the predicted frames and the interpolation ground truth. Thereby, the predicted frames are the result of an interpolation algorithm provided by [3], which is applied to the estimated flow fields. The results of both techniques are given in Table 3.

The first thing we note was that the ranking, based on both average interpolation errors, of *Fusion Flow* is above the ranking of *Segmentation Flow* in seven out of eight test sequences. More precisely, the reversed rank appears at **Basketball**, which contains more motion blur (shutter 16ms) than the other test sequences (shutter 6ms). Its average IE over all region masks is 0.2 graylevels lower than the respective average of *Fusion Flow* (see Table 3). *Segmentation Flow* in this case benefits from its occlusion term. In Figure 2, for example, the white reflection at the door is only visible in one of the two source frames from which the optical flow is computed. While in the interpolation from *Fusion Flow*'s optical flow field this reflection causes an artifact in the reconstruction of the basketball (see Figure 2, left image), *Segmentation Flow* uses the occlusion term to detect and the segment-wise flow to overcome this problem (see Figure 2, right image). On the other hand, its oversmoothed flow field yields visible errors around motion discontinuities. In Figure 2 (right images) this problem can, for example, be observed at the hand on the right, which is visibly split in two parts. The border pixels appear displaced from the hand itself. Algorithms which are capable of preserving sharp motion boundaries, such as *Fusion Flow*, therefore have in comparison to the above mentioned ones, lower measures for the discontinuities region mask for both, the flow error and the interpolation error (see Table 1 and Table 3).

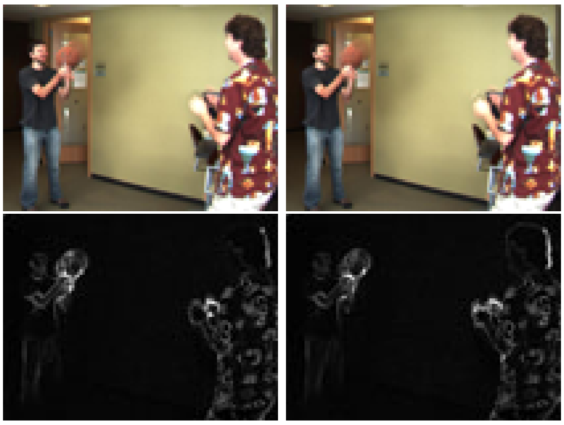

Figure 2: The images above, taken from [3], show predicted frames of **Basketball** (first row) and the corresponding error images (below) for *Fusion Flow* (left) and *Segmentation Flow* (right).

Regarding the errors on Middlebury's interpolation datasets (see Table 3) both, *Fusion Flow* and *Segmentation Flow*, perform better on regularly recorded (e.g. **Schefflera**) than on high-speed datasets (e.g. **Dumptruck**). Regularly recorded sequences with untextured regions, such as **Mequon**, have the advantage that flow errors in these regions do not influence the interpolation errors. However, this set also contains more challenging sequences, such as **Schefflera**. The reason why this sequence is problematic for optical flow estimation techniques or rather frame prediction is its textured, but in terms of color and flow to the foreground similar, background [3]. Therefore, errors, for example caused by this similarity, are visible in the predicted frame and raise the interpolation errors. Still, both algorithms perform well on **Schefflera**. In this context, especially *Fusion Flow* has low interpolation errors (see Table 3). In fact, at the time of comparison only six other submitted approaches (see [3]) have IEs lower than 3.75. The IE of *Segmentation Flow* on this sequence is 0.42 graylevels higher than the respective value of *Fusion Flow*. Its IEs are, in contrast to other datasets, above those of the HS method (see Table 3).

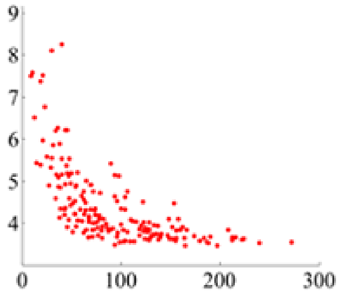

Figure 3: This plot, taken from [1], visualizes the tradeoff between the average AEs (y-axis) and the number of initial solutions (x-axis).

3.3 Runtimes

This section briefly discusses the runtimes of the described estimation techniques. Note that the following numbers are not normalized according programming environment or hardware. Including the computation of 200 initial solutions *Fusion Flow* takes 2666 seconds to process the **Urban** sequence [3]. However, a reduction of initial solutions decreases the runtime without significantly worsening the result (see Figure 3). More precisely, [1] applied the algorithm on a test sequence using subsets of 40 initial solutions. The subset with the minimum AE was only about half a pixel higher than the result with the full set.

Despite of the additional cost of the Mean-Shift segmentation, *Segmentation Flow* processes the **Urban** sequence faster than *Fusion Flow*. It imposes the computational costs of 60 seconds [3].

4 Conclusion

Two stereo inspired methods, *Fusion Flow* [1] and *Segmentation Flow* [2], were presented and compared. Both techniques are top-ranked according to the Middlebury optical flow dataset [3] and cope with common problems in optical flow estimation. Concerning the flow errors, they are ranked above the traditional motion estimation techniques, such as the HS method. We have shown that their results are close to the ground truth, especially for textureless regions. Moreover, *Fusion Flow* (with 200 initial solutions) in summary performs better across the datasets and region masks than *Segmentation Flow*.

The comparison of the error measures demonstrates that the former preserves sharp motion boundaries and copes with non-rigid motion, brightness changes and little contrast between foreground and background. Its weaknesses are the increased error measures for large motion and that it does not handle occlusions. The latter especially matters to applications such as novel view generation, in which occluded areas produce visual artifacts in predicted frames. Additionally, its runtime for 200 initial solutions is 40 times higher than the runtime of *Segmentation Flow*. Reducing the number of initial solutions decreases the runtime, but increases the error measures as well.

Segmentation Flow avoids outliers (e.g. caused by camera noise), benefits from its occlusion term, copes with large motion and is faster than *Fusion Flow*. On the other hand, we have shown that it oversmooths motion boundaries and fine structures. This increases its flow errors and causes visual artifacts, when predicting intermediate frames from the computed flow field. Despite its occlusion term, the oversmoothed results of this method lead to interpolation errors below those of the HS method.

The presented techniques are proving that transfering knowledge from the Stereo Vision literature is beneficial for motion estimation techniques. Both of them still leave room for improvements, but their strengths complement each other. Furthermore, we agree that either incorporating initial solutions from *Segmentation Flow* in *Fusion Flow* or determine pixel-wise *Fusion Flow* in *Segmentation Flow* can yield an enhancement of the respective method in terms of flow and interpolation errors.

References

[1] **Lempitsky V., Roth S., Rother C.**, “FusionFlow: Discrete-continuous optimization for optical flow estimation”. In *CVPR 2008: IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[2] **Xu L., Chen J., Jia J.**, “A segmentation based variational model for accurate optical flow estimation. In *Proceedings of the 10th European Conference on Computer Vision*, Vol. 1, pages 671–684, 2008.

[3] **Backer S., Scharstein D., Lewis J., Roth S., Black M.J., Szeliski R.**, “A database and evaluation methodology for optical flow. In *ICCV 2007: International Conference on Computer Vision*, pages 1–8, 2007.

[4] **Horn B. K. P., Schunck G.**, “Determining optical flow. *Artificial Intelligence*, Vol. 17, Nr. 1–3, pages 185–203, 1981.

[5] **Lucas B. D., Kanade T.**, “An iterative image registration technique with an application to stereo vision. In *IJCAI 1981: International Joint Conferences on Artificial Intelligence*, pages 674–679, 1981.