

LOCAL ALL-PASS FILTERS FOR OPTICAL FLOW ESTIMATION

Christopher Gilliam and Thierry Blu

Department of Electronic Engineering, The Chinese University of Hong Kong
email: {cgilliam, tblu}@ee.cuhk.edu.hk

ABSTRACT

The optical flow is a velocity field that describes the motion of pixels within a sequence (or set) of images. Its estimation plays an important role in areas such as motion compensation, object tracking and image registration. In this paper, we present a novel framework to estimate the optical flow using local all-pass filters. Instead of using the optical flow equation, the framework is based on relating one image to another, on a local level, using an all-pass filter and then extracting the optical flow from the filter. Using this framework, we present a fast novel algorithm for estimating a smoothly varying optical flow, which we term the Local All-Pass (LAP) algorithm. We demonstrate that this algorithm is consistent and accurate, and that it outperforms three state-of-the-art algorithms when estimating constant and smoothly varying flows. We also show initial competitive results for real images.

Index Terms— Optical Flow, All-pass Filters, Approximation, Motion Estimation.

1. INTRODUCTION

An important topic in image processing is the estimation of motion from a sequence of images. This motion is known as the optical flow [1] and is utilised in a wide range of applications such as computer vision, biology [2, 3] and medical imaging [4, 5]. In more detail, first proposed in [1], the problem of estimating the optical flow comprises finding a velocity field u based on the variation of pixel intensities within an image sequence. A standard framework for this problem is to assume a pixel remains constant as it flows from one image to another, this is known as the brightness constraint [6]. Accordingly, two images $I_1(x, y)$ and $I_2(x, y)$ are related as follows

$$I_2(x, y) = I_1(x - u_1(x, y), y - u_2(x, y)), \quad (1)$$

where $u(x, y) = [u_1(x, y), u_2(x, y)]^T$ is the optical flow field and (x, y) is the pixel coordinates. The above constraint is then linearised by performing a first order Taylor approximation under the assumption that the displacement of the flow is small [1, 7]. The result is the standard *Optical Flow Equation*:

$$I_2 - I_1 + u_1 \frac{\partial I_1}{\partial x} + u_2 \frac{\partial I_1}{\partial y} = 0. \quad (2)$$

Note that we have omitted the dependency on x and y for ease of notation. A difficulty associated with (2) however is that it offers one constraint for two unknowns (the aperture problem [8]).

To counter this problem, Horn and Schunck [1] proposed a global approach using L_2 regularization. They minimised the L_2 norm of (2) under the constraint that the optical flow is smoothly varying. Since then, this approach has been extended to use robust penalty terms [9, 10], L_1 regularization [11, 12] and low-rank regularizers [13]. A taxonomy of regularizers was presented in [14]. In

contrast, Lucas and Kanade [7] opted for a local approach. Specifically, they constrained the flow to be constant over a local region and solved (2) within such regions. This local approach is more robust to noise but at the expense of producing a consistent estimate of the flow [10]. To harness both robustness and consistency, the local and global approaches were combined in [10]. Using a similar joint framework, the local constancy constraint on the optical flow has been extended to affine models in [15, 16] and linear filters in [17]. Spectral techniques involving the Radon transform have also been used to estimate multiple superimposed translations in [18] and local affine models in [19]. Finally, feature matching was proposed in [20] and used as an initialisation in [21, 22]. For a complete review of the state-of-the-art see [23, 24, 25, 26, 27], and, more recently, [6, 28].

In this paper, we present a novel framework for optical flow estimation using local all-pass filters. More precisely, instead of assuming small displacement and using (2), we assume the optical flow is slowly changing so that it is locally constant. Under this assumption, we relate local changes between two images via a filter that turns out to be all-pass. Then, we extract the local optical flow from this all-pass filter. Importantly, in contrast to [7], we obtain a consistent estimate of the optical flow. Accordingly, we present a novel filter-based method to estimate smoothly varying optical flows, which we term the Local All-Pass (LAP) algorithm. We evaluate this algorithm in noiseless conditions (i.e. images that exactly satisfy the brightness constraint) and show improved accuracy and speed when compared to state-of-the-art algorithms. We also present initial results for real images taken from the Middlebury website [6].

The paper is organised as follows. In Section 2, we introduce the all-pass filter framework and detail its application when estimating a constant optical flow. Next, in Section 3, we adapt the framework to allow the estimation of a smoothly varying optical flow and present the LAP algorithm. In Section 4, we cover pre- and post-processing techniques that are applied to the LAP algorithm. We then evaluate the LAP algorithm in Section 5 and conclude in the final section.

2. ALL-PASS FILTER FRAMEWORK

The central concept in this framework is that a constant optical flow between two images, I_1 and I_2 , is equivalent to filtering with an all-pass filter H . Using this concept, the framework comprises two-stages: first, we estimate the all-pass filter in question. Then, in the second stage, we extract the optical flow information from the filter. In the following discussion we expand upon these two stages.

2.1. Shifting is all-pass filtering

Assuming the brightness constraint, (1), a constant optical flow is equivalent to shifting an image by a constant displacement vector $u = [u_1, u_2]^T$. Now, using the shifting property of the Fourier

transform, it easily follows that this image shift is equivalent to filtering I_1 . In the continuous domain, the corresponding filter is characterized by a frequency response

$$H(\omega_1, \omega_2) = e^{-ju_1\omega_1 - ju_2\omega_2}.$$

This filter has the following properties

- **Separable:** $H(\omega_1, \omega_2) = H_1(\omega_1)H_2(\omega_2)$;
- **Real:** the impulse response is real-valued and hence $H(\omega_1, \omega_2) = H^*(-\omega_1, -\omega_2)$;
- **All-pass:** $|H(\omega_1, \omega_2)| = 1$.

Now, in the case where sampling is ideal (with sinc-prefiltering), we obtain a digital version of the filter H . Note that the properties of this digital filter are the same as its continuous version. Importantly, the $(2\pi, 2\pi)$ -periodic frequency response of the digital filter can always be expressed as

$$H(\omega_1, \omega_2) = \frac{P(e^{j\omega_1}, e^{j\omega_2})}{P(e^{-j\omega_1}, e^{-j\omega_2})} \quad (3)$$

where $P(z_1, z_2)$ is the 2D z -transform of some separable real filter. A possible choice for $P(e^{j\omega_1}, e^{j\omega_2})$ is $e^{j\frac{1}{2}\arg H_1(\omega_1)} e^{j\frac{1}{2}\arg H_2(\omega_2)}$ using the principal value for the determination of the argument. Note that (3) describes general all-pass filters, not only filters that correspond to a shift.

2.2. Approximating the all-pass filter - A basis representation

Given the frequency response in (3), we propose an approximation approach to obtaining the all-pass filter $H(\omega_1, \omega_2)$. We start by linearising the filter transform been I_1 and I_2 using the all-pass structure in (3) to obtain

$$P(e^{j\omega_1}, e^{j\omega_2})\hat{I}_1(\omega_1, \omega_2) = P(e^{-j\omega_1}, e^{-j\omega_2})\hat{I}_2(\omega_1, \omega_2)$$

where “ $\hat{\cdot}$ ” denotes the Fourier transform of the images. In the sampled space domain, this relation becomes

$$p[k, l] * I_1[k, l] = p[-k, -l] * I_2[k, l], \quad k, l \text{ integer}, \quad (4)$$

where $*$ denotes convolution.

Now, using a standard signal processing technique, we express the filter $P(e^{j\omega_1}, e^{j\omega_2})$ as a linear combination of a few fixed, known real filters $P_n(z_1, z_2)$. In other words, we propose a filter basis representation:

$$P(z_1, z_2) = \sum_{n=0}^{N-1} c_n P_n(z_1, z_2), \quad (5)$$

where N is some small number. As a consequence, our approximation approach to determining $H(\omega_1, \omega_2)$ amounts to finding the coefficients $\{c_n\}_{n=0, \dots, N-1}$ corresponding to the filter basis. Note that by doing so, we will preserve the real all-pass property, but we will lose separability.

A straightforward algorithm for finding the approximation of P consists in minimizing, in the L_2 sense, the difference between the left and right hand sides of (4) when P satisfies the linear representation (5). Algebraically, the result of this minimization is obtained by solving a linear system of equations, which is very fast and efficient (one global minimum only).

2.3. Finding a good filter basis

Having formulated the approximation of the all-pass filter, let us now examine the fixed filters required in (5). One possible approach is to use a canonical filter basis: $P_{m,n}(z_1, z_2) = z_1^{-m} z_2^{-n}$ where $m^2 + n^2 \leq R^2$ and R is the radius of the disk which contains the support of the approximated P filter. This results in a very accurate

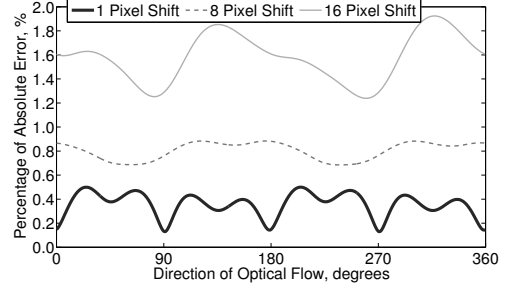


Fig. 1. Graph showing the accuracy of the all-pass framework when estimating a constant optical flow. The accuracy is measured over three different shifts: 1, 8 and 16 pixels, and for 1000 different directions (ranging from 0 to 360 degrees). Note, the accuracy is measured as percentage of absolute error relative to the shift of the flow.

estimation of a constant optical flow; on the order of less than 1% error for a flow with a displacement of 1 pixel. However, such a filter basis requires $N \approx \pi R^2$ different basis elements, which makes it unsuitable for larger displacements: intuitively, R is an upper bound on the displacement of the flow that can be estimated in this context.

In view of this relationship, we propose a compact, yet flexible, filter basis that achieves a similar quality to the canonical basis. Using the Gaussian filter as a template, our basis comprises up to $N = 6$ filters as follows:

$$\begin{cases} p_0[k, l] = e^{-\frac{k^2+l^2}{2\sigma^2}} \\ p_1[k, l] = k p_0[k, l] \\ p_2[k, l] = l p_0[k, l] \\ p_3[k, l] = (k^2 + l^2 - 2\sigma^2) p_0[k, l] \\ p_4[k, l] = kl p_0[k, l] \\ p_5[k, l] = (k^2 - l^2) p_0[k, l] \end{cases} \quad (6)$$

where $\sigma = (R + 2)/4$ and the size of the filters is $(2R + 1)$ by $(2R + 1)$ pixels. Note that, depending on the situation, the number of filters can be reduced in the basis. The advantage of these bases is that they are completely scalable and are typically suited for flows of displacement up to R pixels.

2.4. Extracting the displacement vector

Now, how to retrieve the displacement from the all-pass filter? Since we expect the frequency response of the approximated filter H_{app} to be close to $e^{-j\omega_1 u_1 - j\omega_2 u_2}$, we use the following formula

$$u_{1,2} = j \left. \frac{\partial \log(H_{\text{app}}(e^{j\omega_1}, e^{j\omega_2}))}{\partial \omega_{1,2}} \right|_{\omega_1 = \omega_2 = 0}.$$

This formula has the following simple (and intuitive) expression in terms of the impulse response of the filter P :

$$u_1 = 2 \frac{\sum_{k,l} k p[k, l]}{\sum_{k,l} p[k, l]} \quad \text{and} \quad u_2 = 2 \frac{\sum_{k,l} l p[k, l]}{\sum_{k,l} p[k, l]}. \quad (7)$$

The above expressions proved to be very accurate in all the tests that we have made.

2.5. Examples

To illustrate the accuracy of the all-pass framework, we examine the estimation of three optical flows with constant displacement: 1, 8

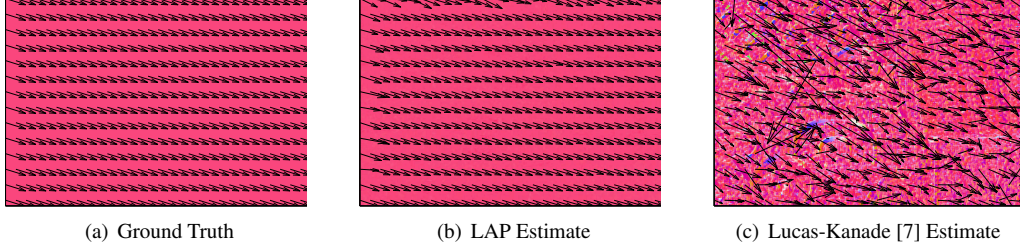


Fig. 2. Graphs comparing the consistency of the raw LAP algorithm (without any pre/post processing) and the Lucas-Kanade algorithm [7] when estimating a constant optical flow. Note that the flow has a displacement of 1 pixel and is applied to the image shown in Fig. 3(a)

Algorithms	Constant Flows				Smoothly Varying Flows				Real Flows			
	D = 1 pixel		D = 15 pixel		D = 1 pixel		D = 15 pixel		Dimetrodon		RubberWhale	
	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE
LAP	4×10^{-6}	1×10^{-7}	0.001	0.001	0.107	0.002	0.746	0.102	1.782	0.096	3.870	0.116
LDOF [21]	0.777	0.020	0.169	0.054	2.119	0.043	11.91	1.310	2.104	0.115	4.310	0.129
MPOF [22]	1.833	0.046	0.094	0.044	2.103	0.041	7.201	0.964	2.976	0.150	2.662	0.087
HS [1, 28]	1.293	0.033	0.084	0.039	1.854	0.037	6.010	0.868	4.562	0.219	3.801	0.119

Table 1. Error comparison for the LAP algorithm and three state-of-the-art optical flow algorithms. In the table, AAE represents the average angular error (in degrees), AEE is the average end-point error (in pixels) and D represents the displacement of the optical flow. The bold values indicate the best result.

and 16 pixels, respectively. For each displacement, we test 1000 different directions of the flow, ranging from 0 to 360 degrees. The results of these estimations are shown Figure 1. The figure shows the absolute error of the estimate as a percentage of the original displacement of the flow. From the figure, we observe that the maximum percentage error is less than 0.5%, 1% and 2% for a shift of 1, 8, 16 pixels, respectively. Note that we use $R = 2D$, where D is displacement, and $N = 6$ filters for this estimation.

3. LOCAL ALL-PASS ALGORITHM

In the preceding discussion, we have assumed the optical flow is constant. We now relax this assumption and consider the estimation of a smoothly varying optical flow. To estimate such a flow, we propose a local adaptation to our framework. Instead of estimating a unique all-pass filter based on the whole image, we assume the flow is constant within a window \mathcal{R} and estimate an all-pass filter within that window. As such, we have the *local all-pass equation* for optical flow estimation:

$$p[k, l] * I_1[k, l] = p[-k, -l] * I_2[k, l], \text{ where } k, l \in \mathcal{R}. \quad (8)$$

Once we have obtained the local filter, we can then shift the window \mathcal{R} and estimate a new local all-pass filter. Based on this concept, we formulate a novel method to estimate a smoothly varying flow, which we term the Local All-Pass (LAP) algorithm. Notice that this is similar to the formulation by Lucas and Kanade [7], however, importantly, we do not use the Optical Flow Equation, (2); thus we are not restricted to flows of small displacement.

More precisely, the LAP algorithm first estimates a local all-pass filter for every pixel in the image - the filter corresponds to the central pixel of the region \mathcal{R} . Then, using these filters, it extracts an estimate of the optical flow according to (7). For a $(2R + 1)$ by $(2R + 1)$ block \mathcal{R} , the estimation stage in the algorithm is equivalent to solving the following minimisation at each pixel

$$\min_{\{c_n\}} \sum_{l, k \in \mathcal{R}} \left| p[k, l] * I_1[k, l] - p[-k, -l] * I_2[k, l] \right|^2 \quad (9)$$

$$\text{where } p[k, l] = p_0[k, l] + \sum_{n=1}^{N-1} c_n p_n[k, l].$$

Notice that we have assumed $c_0 = 1$ thus the above minimisation is equivalent to solving a linear system of equations with $(N - 1)$ unknowns. This solution can be implemented very efficiently using convolutions and pointwise multiplications.

In practice, we may find that the linear system of equations in (9) is singular for certain pixels. However, instead of considering the condition number of the system, we assume these cases are rare and opt to deal with any erroneous estimates that occur in a post-processing stage, see the next section. To validate this approach, Figure 2 highlights the consistency of the LAP algorithm when estimating a constant flow (with a displacement of 1 pixel). The figure compares the raw LAP estimate to that obtained from [7].

4. PRE- AND POST-PROCESSING

The majority of optical flow algorithms employ some form of pre- and post-processing. For example, image pre-filtering is used to enforce the brightness constraint [28] and median filtering is used to improve the optical flow estimate [12]. In our case, we choose to pre-filter real images using a high pass filter based on the Laplacian function. We also incorporate post-processing to aid the accuracy of the LAP algorithm. This post-processing comprises two stages: first, we identify and replace erroneous estimates of the optical flow using an inpainting procedure [29, 30]. We opt for a fast inpainting procedure based on isotropic diffusion [31]. Note that errors are identified in two ways: 1) if they are within $2R$ from the image boundary; 2) if they have a magnitude (i.e. displacement) greater than R . The second stage is to smooth any outlying estimates not previously identified using mean filtering.

4.1. Multi-Scale refinement

Although the LAP algorithm can estimate large optical flows directly, it requires a large filter basis to do so. This is equivalent to assuming large regions of the flow are very slowly varying (i.e. constant). To overcome this issue, we use a multi-scale refinement in which the optical flow is estimated in a coarse-to-fine manner. However, unlike [7, 32], we do not implement this refinement using image pyramids, rather we change the scale of the filters using the

	LAP	LAP w. Median Filters	LDOF [21]	MPOF [22]	HS [1, 28]
Time (seconds)	6.23	7.76	29.87	279.00	47.05

Table 2. Computation time for the five optical flow algorithms. The size of the images used to compute the flow are 388 by 584 pixels.

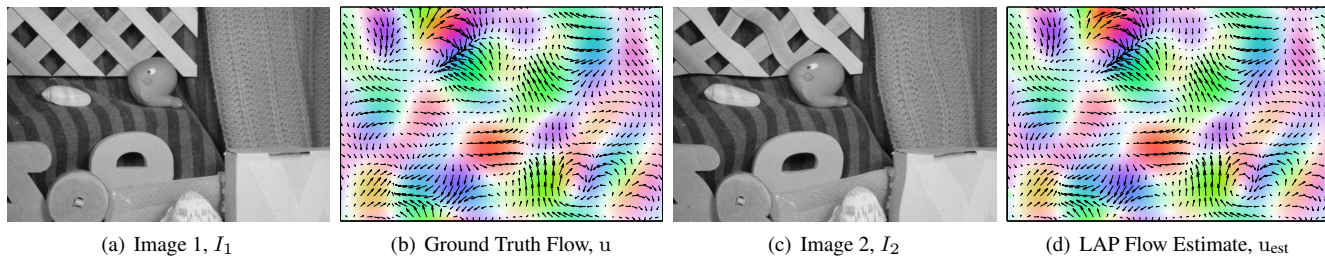


Fig. 3. Illustration of the smoothly varying optical flow and its estimation using the LAP algorithm. The first image is shown in (a), the smoothly varying optical flow in (b), the second image in (c) and the estimate of the flow using the LAP algorithm in (d). Note that the optical flow has a maximum displacement of 15 pixels.

parameter R ; large values of R allow the estimation of large flows whilst small values allow faster variation in the flow. Therefore, we start by estimating a large flow using filters with an equivalent large R and then refine the estimate with smaller values of R .

5. RESULTS

We now compare the performance of the LAP algorithm against two state-of-the-art algorithms: the Large Displacement Optical Flow (LDOF) proposed in [21] and Motion Preserving Optical Flow (MPOF) proposed in [22]. We also use, as a baseline, a modern implementation of Horn and Schunck’s algorithm (HS) presented in [28].

For reference, given the original optical flow u and its estimate u_{est} , performance is measured in terms of the computation time, the End-point Error (EE) and the Angular Error (AE) [25]:

$$\begin{cases} \text{EE} = \|u - u_{\text{est}}\|_2; \\ \text{AE} = \cos^{-1} \left(\frac{1 + u^T u_{\text{est}}}{\sqrt{1 + u^T u} \sqrt{1 + u_{\text{est}}^T u_{\text{est}}}} \right). \end{cases}$$

These errors are then averaged over the whole optical flow. Note that AE is in degrees and only greyscale images are considered.

5.1. Noiseless Conditions

We start by evaluating the LAP algorithm in noiseless conditions. In other words, the second image is generated by warping the first image with the ground truth optical flow. Therefore, the pair of images exactly satisfy the brightness constraint (i.e. they do not suffer from any noise). Under these conditions, we estimate two types of flows: a constant flow and a smoothly varying flow. In both cases, we examine a small displacement, a 1 pixel shift, and a large displacement, a 15 pixel shift. To estimate these flows, we use $N = 3$ in the LAP algorithm and perform 6 multi-scale iterations with the following filter scale values: $R = 32, 16, 8, 4, 2, 2$.

The results of estimating the optical flows with each algorithm are shown in Table 1 and the computation times are shown in Table 2. An illustration of the smoothly varying optical flow (with 15 pixel shift) and its estimation using the LAP algorithm is shown in Figure 3. From the first table, we observe that the LAP algorithm consistently outperforms the other algorithms when estimating the optical flow in noiseless conditions; it is roughly 100 times more accurate for constant flows and about 10 times more accurate for smoothly

varying flows. In terms of computation time, Table 2 shows that the LAP algorithm is the fastest; about 5 times faster than the LDOF, which is the next fastest, and 45 times faster than the MPOF, which is the slowest. Important, unlike the others, this computation time is achieved using only a Matlab implementation. Finally, for the smoothly varying flow with a shift of 15 pixels, we interpolate (using cubic splines [33]) the second image from the LAP, LDOF and MPOF estimates. The resulting PSNR values are 43.9 dB, 28.1 dB and 27.9 dB for the LAP, LDOF and MPOF, respectively.

5.2. Real Images

Now, we present an initial evaluation of the LAP algorithm using two real images (i.e. noisy conditions): the Dimetrodon image pair and RubberWhale image pair from the Middlebury website [6]. For these images, we pre-process them using a Laplacian filter. Also, as real flows are likely to be piecewise, we introduce a two-stage median filtering based on [34]. This filtering is used when $R = 2$ and comprises a coarse 11 by 11 filter followed by a fine 5 by 5 filter.

The results of this initial evaluation are shown in the last column of Table 1. From the table, we observe that the LAP algorithm performs very well on the Dimetrodon images and is competitive on the RubberWhale images. This performance is to be expected as the Dimetrodon flow is approximately smoothly varying while the RubberWhale flow has more edges that are harder for the LAP algorithm. Note that the computation time for the LAP algorithm with the median filtering is shown in Table 2; the inclusion of the median filters increases the time by about 1.5 seconds.

6. CONCLUSION

In this paper, we proposed a new framework for the estimation of the optical flow. The framework is based on using a local all-pass filter to relate a local region in one image to the corresponding region in another image. The optical flow is then extracted from the filter. Using this framework, we presented the novel LAP algorithm to estimate smoothly varying optical flows. It determines a local all-pass filter at each pixel and then extracts the optical flow from these filters. We demonstrated that our algorithm is faster and more accurate than three state-of-the-art algorithms when estimating a constant and smoothly varying flow in ideal conditions. Finally, we presented initial results for real images and showed them to be competitive with the state-of-the-art.

7. REFERENCES

- [1] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, no. 1, pp. 185–203, 1981.
- [2] J. Barron and A. Liptay, "Measuring 3-D plant growth using optical flow," *Bioimaging*, vol. 5, no. 2, pp. 82–86, 1997.
- [3] J. Delpiano, J. Jara, J. Scheer, O. Ramírez, J. Ruiz-del Solar, and S. Härtel, "Performance of optical flow techniques for motion analysis of fluorescent point signals in confocal microscopy," *Mach. Vision Applicat.*, vol. 23, no. 4, pp. 675–689, 2012.
- [4] S. Song and R. Leahy, "Computation of 3D velocity fields from 3D cine CT images of a human heart," *IEEE Trans. Med. Imag.*, vol. 10, no. 3, pp. 295–306, 1991.
- [5] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Trans. Med. Imag.*, vol. 32, no. 7, pp. 1153–1190, 2013.
- [6] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vision*, vol. 92, no. 1, pp. 1–31, 2011.
- [7] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artificial Intell., Vancouver, Canada*, 1981, vol. 2, pp. 674–679.
- [8] E. Hildreth, "Computations underlying the measurement of visual motion," *Artificial Intell.*, vol. 23, no. 3, pp. 309 – 354, 1984.
- [9] M. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Comput. Vision Image Understanding*, vol. 63, no. 1, pp. 75 – 104, 1996.
- [10] A. Bruhn, J. Weickert, and C. Schnorr, "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *Int. J. Comput. Vision*, vol. 61, no. 3, pp. 211–231, 2005.
- [11] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. European Conf. Comput. Vision (ECCV), Prague, Czech Republic*, 2004, vol. 3024, pp. 25–36.
- [12] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for TV-L 1 optical flow," in *Stat. Geometrical Approaches to Visual Motion Anal.*, D. Cremers, B. Rosenhahn, A. Yuille, and F. Schmidt, Eds., vol. 5604 of *Lecture Notes in Computer Science*, pp. 23–45. Springer Berlin Heidelberg, 2009.
- [13] W. Dong, G. Shi, X. Hu, and Y. Ma, "Nonlocal sparse and low-rank regularization for optical flow estimation," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4527–4538, 2014.
- [14] J. Weickert and C. Schnörr, "A theoretical framework for convex regularizers in PDE-based computation of image motion," *Int. J. Comput. Vision*, vol. 45, no. 3, pp. 245–264, 2001.
- [15] S.X. Ju, M. Black, and A. Jepson, "Skin and bones: multi-layer, locally affine, optical flow and regularization with transparency," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition (CVPR), San Francisco, California*, 1996, pp. 307–314.
- [16] T. Nir, A. Bruckstein, and R. Kimmel, "Over-parameterized variational optical flow," *Int. J. Comput. Vision*, vol. 76, no. 2, pp. 205–216, 2008.
- [17] S. Seitz and S. Baker, "Filter flow," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV), Kyoto, Japan*, 2009, pp. 143–150.
- [18] P. Milanfar, "Two-dimensional matched filtering for motion estimation," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 438–444, 1999.
- [19] D. Robinson and P. Milanfar, "Fast local and global projection-based methods for affine motion estimation," *J. Math. Imaging Vis.*, vol. 18, no. 1, pp. 35–54, 2003.
- [20] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.
- [21] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, 2011.
- [22] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, 2012.
- [23] J.K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—a review," *Proc. IEEE*, vol. 76, no. 8, pp. 917–935, 1988.
- [24] M. Otte and H. H. Nagel, "Optical flow estimation: advances and comparisons," in *Proc. European Conf. Comput. Vision (ECCV), Stockholm, Sweden*, 1994, vol. 1, pp. 51–60.
- [25] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [26] A. Mitiche and P. Boutheymy, "Computation and analysis of image motion: a synopsis of current problems and methods," *Int. J. Comput. Vision*, vol. 19, no. 1, pp. 29–55, 1996.
- [27] C. Stiller and J. Konrad, "Estimating motion in image sequences," *IEEE Signal Process. Mag.*, vol. 16, no. 4, pp. 70–91, 1999.
- [28] D. Sun, S. Roth, and M. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *Int. J. Comput. Vision*, vol. 106, no. 2, pp. 115–137, 2014.
- [29] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. Int. Conf. Comput. Graph. Interactive Tech. (SIGGRAPH), New Orleans, Louisiana, USA*, 2000, pp. 417–424.
- [30] A. Bugeau, M. Bertalmio, V. Caselles, and G. Sapiro, "A comprehensive framework for image inpainting," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2634–2645, 2010.
- [31] M. Oliveira, B. Bowen, R. McKenna, and Y. S. Chang, "Fast digital image inpainting," in *Proc. Int. Conf. Visual. Imag. Image Process. (VIIP), Marbella, Spain*, 2001, pp. 261–266.
- [32] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proc. European Conf. Comput. Vision (ECCV), Santa Margherita Ligure, Italy*, 1992, vol. 588, pp. 237–252.
- [33] P. Thévenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 739–758, 2000.
- [34] E. Arias-Castro and D. Donoho, "Does median filtering truly preserve edges better than linear filtering?," *Ann. Stat.*, vol. 37, no. 3, pp. 1172–1206, 2009.