

## Chapter 2

# Optical Flow and Trajectory Methods in Context

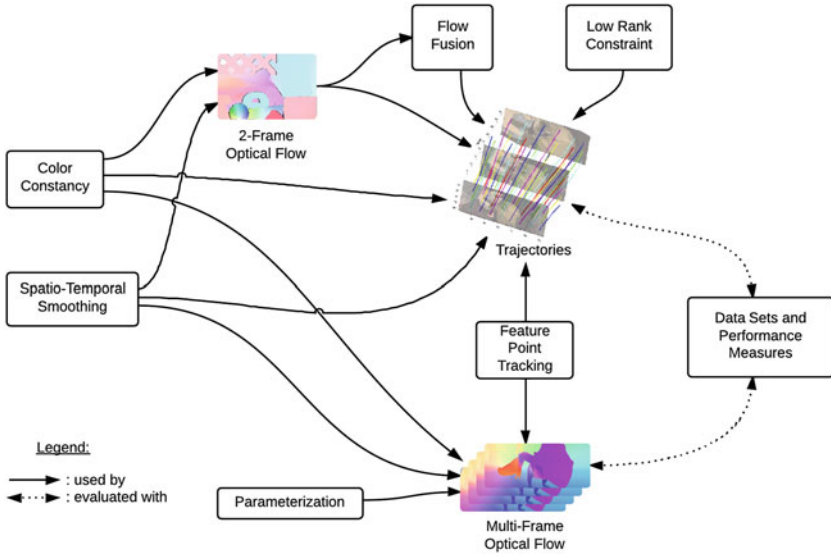
**Abstract** In this chapter we study the related fields of multi-frame optical flow and trajectories. Since the beginning of modern optical flow estimation methods, multiple frames have been used in an effort to improve the motion computation. We look at why most of these efforts have failed. More recently, researchers have stitched together sequences of optical flow fields to create trajectories. These trajectories are temporally coherent, a necessary property for virtually every real-world application of optical flow. New methods compute these trajectories directly using variational methods and low-rank constraints. We also identify the need for appropriate data sets and evaluation methods for this nascent field.

## 2.1 Introduction

Computing motion across a sequence of images is a fundamental computer vision task. Whether an autonomous vehicle seeks to avoid a collision or one is creating a special effect for cinema, identifying the tracks taken across time is crucial. To be useful, a track must be *temporally coherent*, an idea that is not yet achieved by most algorithms currently in use today.

Most optical flow methods are applied to two frames of a sequence in an attempt to map the motion from one frame to the next. These methods are then applied sequentially to pairs of images in a longer sequence. However, ambiguity generally exists because of occlusion, noise, lighting variations, and the general ill-posed nature of the problem. This confusion is easily propagated from one frame to the next creating a cascade of errors. One would rightly expect that looking across multiple frames would provide additional robustness.

Using multiple frames to improve optical flow is not a new idea. In fact, from the beginning of modern optical flow estimation methods, efforts were made to temporally smooth the flow. These early methods were not effective in practice, relying on an accurate locally computed temporal derivative which does not exist in the presence of typical motion. Recently published algorithms show promise but do not yet completely solve the problem, pointing to a huge research opportunity that has yet to be fully exploited.



**Fig. 2.1** Multi-frame optical flow and trajectory methodologies and relationships

This chapter examines two closely related areas, *multi-frame* methods and *trajectories*. Multi-frame methods use a sequence of images to compute one or more optical flow fields. This is a direct extension of optical flow in the temporal dimension. A trajectory, on the other hand, describes the motion of a particle across multiple frames. A dense set of trajectories describes the motion of the scene across a sequence of frames. We do not cover 2-frame optical flow methods. For excellent reviews of this field see Baker et al. [4] and Sun et al. [34].

We postulate that trajectories are the more desirable generalization of optical flow because of their temporal coherence. We also identify opportunities afforded by the lack of recognized trajectory data sets and evaluation methods.

Figure 2.1 shows a graphical abstract of this chapter. It is structured as follows: Sect. 2.2 is the heart of the chapter. It summarizes 30+ references in the field from an algorithmic point of view, reviewing algorithms from the last 30 years and placing them into categories.

Optical flow methods are often evaluated against publicly available benchmarks. Section 2.3 describes the most commonly used benchmarks and performance measures. Section 2.4 discusses the differences between optical flow and trajectories. Finally, Sect. 2.5 presents concluding remarks.

## 2.2 Algorithms

The goal of this section is to provide a representative list of optical flow methods organized by category and put into context. The selected algorithms and methods span 30+ years of research in this field. The categories here are more inspired by the chronological development of ideas rather than any mutually exclusive partition. There is an unavoidable overlap between the categories. For example, spatio-temporal smoothing is present in parameterization methods. Low-rank methods can be seen as a parameterization but are placed in their own section; one fusion flow method appears in the low rank category. When multiple methods are combined the most prominent or novel feature is used.

### 2.2.1 *Spatio-Temporal Smoothing*

There were many efforts to expand 2-frame optical flow estimation methods into multiple frames in the two decades following its “birth” with the parallel works of Horn and Schunck [18] and Lucas and Kanade [21] both in 1981. Since spatial gradients were being used to guide the flow in the 2-frame case there was a natural expansion into 3-D gradients of the spatio-temporal volume consisting of a sequence of frames stacked together.

Murray and Buxton [22] were concerned with solving segmentation using motion cues. They assume that the 3D scene they are considering is comprised of locally planar facets which greatly simplifies their motion model. They use an MRF to enforce a piecewise smooth spatial-temporal regularization and solve with simulated annealing. They assume that motion is temporally constant. This assumption, while a reasonable starting point is not supported in natural image sequences.

Heeger [17] proposes using 3D Gabor filters operating on a spatio-temporal volume. He operates on an image pyramid to find the scale which best matches the motion amount to his filters. Because the filters are relatively large, the results are poor with smaller objects or fine structure. Furthermore, the results are not valid near motion boundaries.

Nagel [23] proposes an extension of his image-driven smoothing across discontinuities into the spatio-temporal realm. He does not implement or show results but presents a theoretical framework to do so.

Singh [33], Chin et al. [11], and Elad and Feuer [13] all used a Kalman filter to enforce smoothness in the temporal domain.

Weickert and Schnörr [38] derive a flow-driven spatio-temporal smoothness constraint. Using a linearized color constancy in their objective, they solve the Euler-Lagrange equations.

Bruhn et al. [9] proposed a spatio-temporal version of their *Combined Local Global* algorithm. This algorithm combines the global nature of Horn and Schunck [18] with the local area matching of Lucas and Kanade [21].

Salgado and Sánchez [30] separate the spatial and temporal smoothing. Spatially they follow Nagel’s [23] image-driven smoothing. Temporally they penalize any change in flow across frames. They solve Euler-Lagrange equations across a pyramid of images. Using two synthetic sequences with simple motion, they demonstrate improvement with their temporal method over spatial smoothing alone. According to [36], this temporal smoothing fails by oversmoothing complex motion.

Zimmer et al. [39] use a linearized version of the color constancy and then solve the Euler-Lagrange equations. For data constancy they use color and gradient of each HSV color channel each with a robust  $\ell_1$  approximate penalty. They point out limitations of image-driven regularization results in oversegmentation because every image edge does not correspond to a flow edge. They note flow-driven regularization does not give as clean flow edges as image-driven results. They combine the two using  $\|G\nabla u\|_1$  where  $G$  is tensor-directed based on image gradient directions (image-driven) and the  $\ell_1$  norm is the flow-driven regularization. They observe that Middlebury sequences have too large of a displacement hence the assumption of a temporally smooth field is violated. So they use the *Marble* sequence with its slow motion to show spatio-temporal improvement over the 2-frame case.

**Summary** We observe that using a temporal smoothness as a constraint shows improvement over 2-frame methods for sequences with small frame-to-frame displacements. Unfortunately, for larger displacement, the locally generated temporal derivatives do not agree with the actual motion. This failure mode is in fact ordinary rather than the exception. Temporal sampling is far lower than the Shannon sampling rate and typical motion in fact is highly aliased. While spatial resolution is inherently bandlimited by camera optics, there is no natural mechanism which limits physical motion relative to the camera frame rate. This stumbling block has stifled development in this direction.

### 2.2.2 *Parameterizations*

In this subsection we present methods that make specific assumptions about the motion or camera model, then use this to predefine some basis functions.

Black and Anandan [6] assume that the image plane acceleration of a patch is approximately constant temporally and they construct a running average to constrain the variation with time. They further use a robust function to spatially smooth the flow while rejecting outliers, an improvement over the quadratic penalty which was commonly used at the time.

Black [5] improves his previous work to include a robust penalty on the constant acceleration assumption. Furthermore, he solves the problem with a continuation method called *Graduated Non-convexity* as a more deterministic strategy than the simulated annealing.

Volz et al. [36] uses a 5 frame sequence with a variational formulation. Their data constancy term separately penalizes the 4 incremental flows. Their objective functions sums the color constancy error across all 3 color channels. They also use a

tensor-directed spatial regularizer. They experiment with applying the regularizer on each of the 4 flow fields separately and summing versus applying the tensor-directed regularizer to each flow field summing the result then applying an  $\ell_1$  approximation. The joint regularization seemed to generally work better than penalizing each separately.

They try to fit trajectories to a parabolic curve. This is based on their assumption that the projected 2D trajectories first derivatives are continuous. They compare just spatial regularization with 1st and 2nd order trajectory regularization to a locally adaptive version and a globally adaptive version. There was no clear winner. The parabolic trajectory assumption is similar to the constant acceleration assumption of [6].

They solve the Euler-Lagrange equations using a Gauss-Seidel multigrid solver in a coarse to fine framework. They make the point that they do not use a coarse-to-fine linearization of the problem or image warping. Because of this they outperform methods using a linear color constancy on large displacement motion. They instead linearize in the solution space via their multi-grid method. This involves “W” cycles of moving from coarse to fine several times to try to find a global minimum to a non-convex problem.

Nir et al. [24] use an over-parameterized space-time model to describe flow. Total variation regularization is applied to the coefficients of the basis functions. The basis functions are chosen to fit either an affine model, rigid-body motion, and special cases such as translational and constant motion. Euler-Lagrange equations are solved in a multi-scale framework. They showed improvements on very simple motion examples, *Yosemite*, *Flower Garden*, and a synthetic translational sequence. The basis functions consists of up to quadratic coordinate terms of pixel location chosen to fit the particular model. The spatio-temporal results were only reported on Yosemite.

**Summary** Simplifying the motion model allows a parameterization of the flow or trajectory. If the basis chosen can accurately represent the motion in the scene then the ill-posed flow problem is constrained, thereby improving results. Unfortunately, for general motion these models do not hold, limiting their usefulness to special cases.

### 2.2.3 Optical Flow Fusion

In this section we examine methods that begin with a sequence of 2-frame optical flow fields then fuse matching tracks into longer trajectories.

Sand and Teller [31] use a combination of sparse point tracking which extends over multiple frames and dense optical flow which is computed across two frames which they call *particle video*. The big difference between particle video and a dense trajectory is that the particle density is adaptive based on the motion detail so that fewer points have to be tracked.

Sand and Teller start with a KLT-based global motion stabilization step. They then compute 2-frame optical flow estimates for all the frames in the sequence. They identify occluded pixels and then mask the pixels from the color constancy error. They dryly point out that robust regularizers produce better looking transitions that are still wrong. They use Delaunay triangulation to link adjacent particles. Weights of edges are computed based on similarity of motion. Particles moving similarly will have strong links while links will be nearly zero across occlusion boundaries. Particles that have high distortion errors, a measure of how dissimilar their trajectory is to their neighbors, get pruned. Pixels whose color is scale invariant can get added as a particle to fill out areas of lower density. The number of particles tracked are much smaller than the number of pixels. This shorthand description of the image motion is computationally advantageous for matting, compositing and similar tasks. Sand and Teller’s distortion measure and pruning can be seen as similar to the low rank DCT basis used later by Garg et al. [15], i.e. the assumption of high correlation between trajectories.

Rubenstein et al. [29] point out flaws in Particle Video (PV) [31] where frames far apart match a particle of a totally different color. Their work aims at achieving long motion trajectories and is not spatially dense. Interestingly, they use KLT as an initialization because they found it more stable than *Particle Video* or Brox and Malik’s *Large Displacement Optical Flow* (LDOF) [8]. KLT continually drops tracks and restarts new ones with time. These authors use an MRF formulation to link temporally separated tracks. Along with a track compatibility measure they impose a track regularization which avoids crossing trajectories behind occlusions. Claiming no real benchmark for long term tracking existed they created some synthetic sequences. They show their methodology would improve PV and LDOF when they were used as initializers. They demonstrate the usefulness of these sparse long trajectories in human-action recognition problems.

Crivelli et al. [12] compute long-range motion by an iterative approach that works backwards from the reference frame  $N$  which is the last frame of the considered sequence. They look for the best path forward from the current frame  $n$  to frame  $N$ . They may choose from all the paths in frames between  $n, \dots, N$ , to pick up a trail that may have been lost. In this way they are not forced to find a match in frame  $n + 1$  when the match there may be occluded. They expand their backtracking approach to the symmetrical idea of considering frames after the reference frame which in turn is iteratively computed forwards from backwards-pointing flow. They use a Potts-style regularizer and solve for their optimal solution via graph cuts. They demonstrate some visual improvement over Sundaram and Brox [35].

**Summary** These flow fusion methods were the first to construct long trajectories while considering occlusion. Their emphasis is on finding the best way to connect disparate flow fields. Their results are limited by the quality of the flow field inputs.

### 2.2.4 *Sparse Tracking to Dense Flow*

Long term tracking of sparse points even with large displacement is a well developed topic, epitomized by the famous KLT tracker [32]. These trackers can often track objects across many frames although feature points get dropped when they can no longer be reliably matched and new ones are added. The algorithms in this subsection use feature point matching to try to address the large motion weakness of coarse-to-fine flow methods.

Brox and Malik [8] capture large motion using SIFT or HOG descriptors. This is then incorporated into a nonconvex variational model. They formulate the flow objective with a nonlinear optical flow constraint, using color and gradient constancy. The feature point matching flow is also used as a constraint but its influence is reduced to zero as the image is taken coarse to fine. Euler-Lagrange equations are then solved. Middlebury *Average Angular Error* is reported but the rest is qualitative comparison with other methods.

Sundaram et al. [35] is a GPU implementation of [8]. There are some GPU-specific adaptations from [8] including HOG descriptors and Conjugate Gradient solver with a pre-conditioner. They evaluate the second eigenvalue of the image structure tensor and discard any tracking points with the eigenvalue smaller than a threshold. Tracks are stopped when forward and backwards flow do not match. Additionally, tracks that are get very close to flow boundaries are stopped to prevent inadvertent drift to the other side of the boundary. The authors show a 66 % improvement against Particle Video [31] using Sand and Teller's data set. It also outperforms a KLT tracker in terms of accuracy and density. Because of the GPU implementation it also runs  $78\times$  faster than the CPU version.

Lang et al. [20] construct a method quite different from the other variational methods presented here. They pose the optical flow problem as a heat equation with discrete SIFT point matching becoming Dirichlet boundary conditions. By posing spatial smoothing as a quadratic penalty of the flow gradient à la Horn and Schunck, they recall that the solution is equivalent to Gaussian convolution. They apply this spatial filter within an edge-aware domain transform. Temporally they use a simple box filter along tracks. They trace a path or track forward from a reference frame until the track leaves the image or multiple tracks converge on one pixel. For pixels that do not have a track, a new one is started. In the case of multiple tracks on a single pixel, one is discarded at its previous frame. These techniques produce a temporally consistent result. Spatially, however, their smoothing method seems vulnerable to objects that lack sharp gradient boundaries to constrain the smoothing. Their positive results are mostly qualitative visual comparisons. The separable nature of the Gaussian filter enables their algorithm to execute quickly, about  $64\times$  faster than Volz et al. [36].

**Summary** By choosing tracking points that are distinctive across multiple frames, this strategy combines some of the best of two worlds. The high-confidence discrete point tracks are used to constrain variational optical flow to produce a dense flow field.

### 2.2.5 Low Rank Constraints

Under certain camera models or types of motion, the trajectories of a scene form a low rank matrix. We look at the research that leverages this property. Low rank trajectories constraint is not an extension of 2-frame optical flow but is only meaningful in the context of trajectories.

To illustrate a typical trajectory representation used by work in this section: Let  $N$  be the number of pixels in a frame and  $F$  be the number of frames in a sequence. Let  $u, v$  be the previously defined horizontal and vertical optical flow between a reference frame and every other frame in  $1 \dots F$ . Here the flow fields have been vectorized and written as rows of the matrix

$$\mathcal{U} \triangleq \begin{bmatrix} u_{11} & \cdots & u_{1N} \\ \vdots & & \vdots \\ u_{F1} & \cdots & u_{FN} \\ v_{11} & \cdots & v_{1N} \\ \vdots & & \vdots \\ v_{F1} & \cdots & v_{FN} \end{bmatrix}. \quad (2.1)$$

The  $i$ th column of this matrix represents the trajectory of the  $i$ th pixel in the reference frame.

Brand [7] infers a 3D non-rigid model and its motion from a single video camera with a weak perspective camera assumption.

Irani [19] considers camera and motion models that result in a low rank trajectory matrix. Her work is derived from work in Structure from Motion (SfM) but she does not do any 3D reconstruction in this work. Further, she computes a dense set of trajectories for each pixel in a reference frame as in Eq. 2.1. This is different from most SfM work which requires a set of matched points that appears in each frame. She uses the posterior inverse covariance matrix of the flow as a confidence measure. These confidence measures ensure the low rank flow computed is propagated to regions of low confidence rather than allowing the low confidence regions to corrupt high confidence areas. Specifically, Irani shows how this low rank constraint applies to linearized camera models, such as orthographic, weak-perspective, and to full perspective cameras undergoing small rotation and forward translation.

Garg et al. [14] use “reliable” tracks to compute a trajectory basis. They use an  $L^2$  norm for color constancy and spatial smoothing. They use sequences about 50 frames long with only 5 basis elements. That is to say the column space of Eq. 2.1 is spanned by these 5 vectors. They parameterize the trajectories with these basis functions. In a later paper [15] they called that a hard constraint compared to their later soft constraint. They form the Euler-Lagrange equations and use a coarse-to-fine framework with linearization and warping to create linear equations solved with the SOR solver.



Ricco and Tomasi’s *Lagrangian Motion Estimation* (LME) [25] combine occlusion and low rank multi-frame trajectories. They first use a standard tracker on key points and derive a trajectory basis from these. Trajectories in general are not of low rank but they claim the rank is often small in practice. They use the first and last frame of a sequence as reference frames. They reason that using the first frame is fine for points that are not occluded there but useless otherwise. Clearly, using only first and last does not solve this problem but mitigates it somewhat. They use a nonlinear color constancy and an image-driven TV regularizer. They have an occlusion mask variable  $\nu$  which they call a visibility flag. They apply two energy functions to  $\nu$ , the first is an  $L^2$  penalty for  $\nu$  differences from  $\nu^0$ , a starting per-trajectory initialization. Secondly, they enforce a TV smoothness constraint which is relaxed when the trajectories change direction or the second temporal derivative of the brightness constancy error is high. The visibility flag allow entries in  $\mathcal{U}$  which are low-rank constrained but do not contribute to the color constancy error. They only declare a trajectory occluded if they can find the occluding track. They visually outperform LDOF for trajectories that track heavily occluded areas, e.g. a lady walking in front of scene.

Garg et al. [15] implement a low-rank trajectory soft constraint with an anisotropic regularization. This is implemented in CUDA on an NVIDIA GPU. They created a trajectory benchmark based on a synthetic flag waving sequence. They outperformed LDOF and ITV- $L^1$  by Wedel et al. [37]. Their objective function contains a Huber- $L^1$  robust color constancy term with a Huber- $L^1$  approximation of Total Variation of the basis coefficients. They have an  $L^2$  soft coupling term between their color constancy trajectory and the low-rank generated flow  $\mathcal{U}$ . They show the soft coupling produces better results than their previous “hard coupling”. This can somewhat be explained since they do not use any occlusion masking. Results are shown for PCA basis trajectories computed from KLT tracks as well as a pre-defined DCT basis. The PCA performs better all although not always by a lot. One may consider these trajectories as a best fit low rank regression to the frames where the points are visible. In one way this is a reasonable approximation of the physical 3D path. The physical trajectory does not vanish just because it is not visible in a frame or two.

Ricco and Tomasi [27] significantly refine their previous model tying together low-rank trajectories and occlusions. Their objective consists of a data constancy and a temporal smoothing term. The data term is an  $L^1$  approximation of a nonlinear color constancy. Data constancy is masked in occluded areas. Temporal smoothing is achieved by constraining trajectory coefficients that describe nearby paths through similar appearance to have similar values. Their visibility flag  $\nu$  is formulated as a Markov Random Field and solved concurrently with the trajectories using graph cuts.

Initially, KLT sparse tracking is performed. They apply a two-frame optical flow method to sequential pairs to assemble tracks similar to Sundaram et al. [35]. These tracks are used to find a basis. This process is described by Ricco and Tomasi [26].

Anchor points start in the first frame, for tracks that begin there. For tracks that end in last frame, anchor points are placed there. For tracks that begin or end at frames in between, anchor points may be placed in those frames, particularly for areas that

are occluded in the first or last frame. Anchor points are not placed in a thin barrier around motion boundaries.

Optimization alternates between solving a continuous optimization for the trajectory coefficients given a fixed set of visibility flags. Then with trajectory fixed the combinatorial visibility flags problem is solved via graph cuts. Anchor points are added until each of their pixels have a track within one pixel and invisible trajectories are removed. Optimization stops when trajectories change less than a pixel in every frame.

From their discussion, it is implied that they perform poorly on highly deformable objects like Garg’s flag sequence and on crowds. Their model is susceptible to lighting changes over these long sequences. They log a computation time of 150 hours for 60 frames on the *marple1* sequence.

They outperform their previous LME results as well as LDOF. In the absence of a benchmark, they measure how closely a trajectory maintains color constancy. They also measure mean path length and pixel distance to nearest track. They dramatically beat LDOF on the pixel distance and soundly improve on their previous LME method. Oddly, they lose on path length to LME on every sequence!

Ricco points out in [28] that MPI-Sintel’s data set comes closest to being useful with their longer sequences and ground truth optical flow for each frame but argues this is still not useful for long trajectories since they do not follow any long term correspondences.

**Summary** Low rank constraint is undoubtedly a powerful tool for orthographic camera models or weak perspective models. One researcher applied the low rank constraint to more general cases and found that it can be helpful. However, a full perspective camera with large z-motion can produce trajectories that are far from low rank. So the usefulness of this technique for general motion is unclear.

## 2.3 Data Sets and Performance Measurement

### 2.3.1 Existing Data Sets

The goals of this section are to remind the reader (1) This field is highly driven by comparing new methods against publicly available popular benchmarks. (2) Middlebury [3, 4] is by far the most popular of the benchmarks. (3) Middlebury is a combination of sequences, ground truths, and performance measures. Despite its popularity it is not suitable for multi-frame for these reasons:

1. The Middlebury evaluation set has up to eight frames in a sequence but provides a ground truth for only the center pair.
2. For the low rank methods, eight frames form too short of a sequence to add a meaningful constraint.
3. No trajectory ground truth.

To fill this gap different data sets and performance measures have been proposed. This section discusses some of them. Middlebury, while venerable, is now considered largely solved and lacking challenges of realistic sequences. The most prominent new contenders are the KITTI benchmark and evaluation [2, 16] and the MPI-Sintel dataset [1, 10]. Interestingly, top performers on Middlebury have been seen to not perform well on KITTI and MPI-Sintel. KITTI offers a multiview version of its data set, with 20 frames per sequence. Ground truth however, is provided for the center frame only. The MPI-Sintel data set is the most promising, providing sequences 20–50 frames long with 2-frame optical flow ground truth between each pair of frames.

There are almost no publicly available data sets for trajectory evaluation. This has led to each research effort choosing different sequences and metrics for evaluation. This makes it difficult to compare the performance of different approaches.

For 2-frame optical flow measurement methods see [4]. These metrics have been applied with multi-frame methods which generate 2-frame flow results. In broad strokes, they provide three measures. The Average Endpoint Error and Average Angular error provide two ways of comparing the average difference of a field of 2-D vectors. According to Baker et al. [4], the Average Endpoint Error is the preferred metric of the two. The third metric is specific to image sequences. Suppose we capture a series of high speed photographs. If we compute optical flow correctly between a reference image and the second image following, the flow should be capable of accurately predicting the intermediate image. Since the intermediate image is known we compute an RMS error between our interpolated frame and the captured ground truth. For difficult sequences a visual qualitative inspection is more informative than numbers alone. An extended version of this methodology is often used with trajectories to warp sequences back to a reference frame.

### 2.3.2 *Individual Measurement Efforts*

Sand and Teller [31] took video sequences and appended the reverse order of the same frames. They admit this should not be used for future comparisons. The only known here is that the last frame is the same as the first, so the zero vector satisfies this requirement without following any meaningful trajectory in the intermediate frames.

For qualitative comparison Ricco and Tomasi [27] use the trajectories to warp all frames to a reference frame. Quantitatively, they measure color constancy along a trajectory masked by occlusion. To avoid gaming this admittedly weak metric, they also measure the average visible length of the tracks. While color constancy and long visible tracks are desirable traits, neither directly imply accuracy. They also measure the distance from every pixel in the sequence to the nearest visible track to demonstrate the density of their implementation. However their implementation adds tracks until each of their pixels have a track within a one pixel.

Garg et al. [15] warps real-life sequences and add textures to show qualitative improvement. For quantitative improvement they use a motion captured flag sequence

to create an orthographic camera synthesized version with ground truth. They have made this available on their web site. The sequences used seem to be selected such that trajectories do not leave the raster.

As trajectory algorithms improve, sequences with a perspective camera model and with content entering and leaving the image will be necessary.

**Summary** The path to recognition for any new 2-frame optical flow method runs through one of the publicly available evaluation sites. An opportunity awaits for a similarly useful data set and evaluation site for trajectory algorithms.

## 2.4 Trajectory Versus Flow

In this section we argue that trajectories are fundamentally different, more useful, hence more valuable, than sequences of 2-frame flow fields.

Multi-frame methods offer the possibility of improving 2-frame flow methods by applying additional constraints. Trajectory representation brings a temporally coherent motion that is desirable in most applications and necessary in others.

The 2-frame flow methodology grew as a 2-D extension of the 1-D stereo depth problem. Indeed, the Middlebury optical flow set contains stereo pairs as 1-D flow sets. However, unlike stereo data sets where additional cameras will not be added *ex post facto*, the premise of video is the existence of additional frames. It is difficult to think of an uncontrived application where optical flow computation stops after two frames. Perhaps the strongest legacy reason for 2-frame methods is their smaller computational requirements, in particular smaller memory.

Because of the ill-posed nature of optical flow, any method is presumed to contain errors. Simply concatenating these errors across frames quickly results in wildly inaccurate trajectories. This leaves the application relying on the flow, to somehow smooth the temporal inconsistencies. This effort, clearly is better conducted in the environment where flow constraints are present and can be managed.

To illustrate the importance of temporal consistency we consider a few ordinary applications of optical flow.

Optical flow is used in media and entertainment applications to add special effects, compositing, de-interlacing, speed changes, denoising and restoration. Since the end result is video or a cinema sequence, there must be temporal coherence. It is easier to tolerate small, temporally consistent errors here than similarly small uncorrelated errors which appear as frame-to-frame jittery noise.

Optical flow can be used to help disambiguate other inverse computer vision problems such as stereo depth (scene flow), segmentation, denoising, and structure from motion.

In robotic applications optical flow provides feedback on ego-motion as well as environmental hazards. Indeed, the KITTI evaluation benchmarks are street scenes captured from a moving vehicle. The requirements here for real-time, low-latency feedback provide a constraint on the number of look-ahead frames that are useful for a multi-frame scheme.

A sequence of two frame optical flow fields does have an advantage over a trajectory when warping an image. That is, optical flow provides a map from the center of each pixel, ignoring occlusion. Trajectories, on the other hand, may be anchored on a reference pixel in one frame and miss pixel centers in all the others. Of course, the value of the flow at a pixel center can be interpolated, or conversely with dense trajectories the color at the trajectory center can be interpolated.

It would be useful to convert trajectories to 2-frame flows or some intermediate length. This would ease comparisons not addressed by trajectory benchmarks. Additionally, it is convenient in image warping to have a flow or trajectory that is referenced to that image pixel centers. While methods have been introduced to fuse short flows to longer trajectories, the opposite has not been studied.

**Summary** Applications that use optical flow are ultimately interested in estimating the temporal consistency of trajectories. Trajectory computation is typically more expensive than 2-frame flow fields. Faster computers and better optimization methods are enabling technologies to the implementation of emerging trajectory estimation methods.

## 2.5 Conclusions

Since near the beginning of modern optical flow estimation methods, there have been efforts aiming at improving flow results with knowledge from additional frames. The first two decades of research focused on methods using the local computation of temporal derivatives. We have discussed how this endeavor fails in general. Research in the last 15 years has moved beyond color constancy to use low rank constraints and optimized bases to describe trajectories. We argue that 2-frame optical flow is largely a stop-gap for trajectories with their inherent temporal consistency. Algorithm development for dense trajectories is still immature. Similarly, the field lacks a recognized data set and evaluation methodology. We expect to see development along these three avenues — algorithms, data sets, and evaluation methods — to be further advanced in the next few years.

## References

1. MPI Sintel Flow Dataset. <http://sintel.is.tue.mpg.de/> (2014)
2. The KITTI Vision Benchmark Suite. <http://www.cvlibs.net/datasets/kitti/index.php> (2014)
3. The Middlebury Computer Vision Pages. <http://vision.middlebury.edu> (2014)
4. S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, R. Szeliski, A database and evaluation methodology for optical flow. *Int. J. Comput. Vis.* **92**(1), 1–31 (2011)
5. M. Black, Recursive non-linear estimation of discontinuous flow fields, in *Computer Vision—ECCV’94*, pp. 138–145 (1994)
6. M.J. Black, P. Anandan, Robust dynamic motion estimation over time, in *Proceedings of Computer Vision and Pattern Recognition, CVPR-91*, pp. 296–302 (1991)

7. M. Brand, Morphable 3D models from video, in *Computer Vision and Pattern Recognition*, pp. II-456–II-463, vol. 2 (2001)
8. T. Brox, J. Malik, Large Displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(3), 500–513 (2011)
9. A. Bruhn, J. Weickert, C. Schnörr, Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vis.* **61**(3), 211–231 (2005)
10. D.J. Butler, J. Wulff, G.B. Stanley, M.J. Black, A naturalistic open source movie for optical flow evaluation, in *European Conference on Computer Vision (ECCV)*, pp. 611–625 (2012)
11. T.M. Chin, W.C. Karl, A.S. Willsky, Probabilistic and sequential computation of optical flow using temporal coherence. *IEEE Trans. Image Process.* (A Publication of the IEEE Signal Processing Society). **3**(6):773–788 (1994)
12. T. Crivelli, P. Conze, P. Robert, P. Pérez, From optical flow to dense long term correspondences, in *International Conference on Image Processing (ICIP)*, pp. 61–64 (2012)
13. M. Elad, A. Feuer, Recursive optical flow estimation adaptive filtering approach. *J. Vis. Commun. Image Represent.* **9**(2), 119–138 (1998)
14. R. Garg, L. Pizarro, D. Rueckert, L. Agapito, Dense multi-frame optic flow for non-rigid objects using subspace constraints. *Comput. Vis. ACCV* **2010**, 460–473 (2011)
15. R. Garg, A. Roussos, L. Agapito, A variational approach to video registration with subspace constraints. *Int. J. Comput. Vis.* **104**, 286–314 (2013)
16. A. Geiger, P. Lenz, and R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in *Conference on Computer Vision and Pattern Recognition (CVPR)* (2012)
17. D. Heeger, Optical flow using spatiotemporal filters. *Int. J. Comput. Vis.* 279–302 (1988)
18. B. Horn, B. Schunck, Determining optical flow. *Artif. Intell.* **17**, 185–203 (1981)
19. M. Irani, Multi-frame correspondence estimation using subspace constraints. *Int. J. Comput. Vis.* **48**(153), 173–194 (2002)
20. M. Lang, O. Wang, T. Aydin, Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph.* **31**(34):31:4–31:8 (2012)
21. B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in *Proceedings of Imaging Understanding Workshop*, pp. 121–130 (1981)
22. D. Murray, B.F. Buxton, Scene segmentation from visual motion using global optimization. *Pattern Anal. Mach. Intell.* *IEEE Trans. PAMI.* **9**, 220–228 (1987)
23. H.H. Nagel, Extending the ‘oriented smoothness constraint’ into the temporal domain and the estimation of derivatives of optical flow, in *Proceedings of the First European Conference on Computer Vision*. (Springer New York, Inc., 1990), pp. 139–148
24. T. Nir, A.M. Bruckstein, R. Kimmel, Over-parameterized variational optical flow. *Int. J. Comput. Vis.* **76**(2), 205–216 (2007)
25. S. Ricco, C. Tomasi, Dense lagrangian motion estimation with occlusions, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1800–1807 (2012)
26. S. Ricco, C. Tomasi, Simultaneous compaction and factorization of sparse image motion matrices. *Comput. Vis. ECCV* (2012)
27. S. Ricco, C. Tomasi, Video motion for every visible point, in *International Conference on Computer Vision (ICCV)*, number i (2013)
28. S.M. Ricco, Video motion: finding complete motion paths for every visible point, in *ProQuest Dissertations and Theses*, p. 142 (2013)
29. M. Rubinstein, C. Liu, W. Freeman, Towards longer long-range motion trajectories, in *Proceedings of the British Machine Vision Conference* (2012)
30. A. Salgado, J. Sánchez, Temporal constraints in large optical flow estimation, in *Computer Aided Systems Theory EUROCAST 2007*, vol. 4739, Lecture Notes in Computer Science, ed. by R. Moreno-Díaz, F. Pichler, A. Quesada-Arencibia (Springer, Berlin, 2007), pp. 709–716
31. P. Sand, S. Teller, Particle video: long-range motion estimation using point trajectories. *Int. J. Comput. Vis.* **80**(1), 72–91 (2008)
32. J. Shi, C. Tomasi, Good features to track, in *IEEE Conference on Computer Vision and Pattern Recognition 1994* (1994)

33. A. Singh, Incremental estimation of image-flow using a kalman filter, in *Proceedings of the IEEE Workshop on Visual Motion*, pp. 36–43 (1991)
34. D. Sun, S. Roth, M.J. Black, A quantitative analysis of current practices in optical flow estimation and the principles behind them. *Int. J. Comput. Vis.* **106**(2), 115–137 (2013)
35. N. Sundaram, T. Brox, K. Keutzer, Dense point trajectories by GPU-accelerated large displacement optical flow, in *Computer Vision ECCV 2010* (Springer Berlin Heidelberg, 2010)
36. S. Volz, A. Bruhn, L. Valgaerts, H. Zimmer, Modeling temporal coherence for optical flow, in *2011 International Conference on Computer Vision (ICCV)*, pp. 1116–1123 (2011)
37. A. Wedel, T. Pock, C. Zach, H. Bischof, D. Cremers, An improved algorithm for TV-L 1 optical flow, in *Statistical and Geometrical Approaches to Visual Motion Analysis*, pp. 23–45 (Springer Berlin, Heidelberg, 2009)
38. J. Weickert, C. Schnörr, Variational optic flow computation with a spatio-temporal smoothness constraint. *J. Math. Imaging Vis.* pp. 245–255 (2001)
39. H. Zimmer, A. Bruhn, J. Weickert, Optic flow in harmony. *Int. J. Comput. Vis.* **93**(3), 368–388 (2011)

Optical Flow and Trajectory Estimation Methods

Gibson, J.; Marques, O.

2016, X, 49 p. 6 illus., Softcover

ISBN: 978-3-319-44940-1