Dense Lagrangian Motion Estimation with Occlusions

Susanna Ricco Carlo Tomasi
Duke University
Durham, NC USA
{sricco, tomasi}@cs.duke.edu

Abstract

We couple occlusion modeling and multi-frame motion estimation to compute dense, temporally extended point trajectories in video with significant occlusions. Our approach combines robust spatial regularization with spatially and temporally global occlusion labeling in a variational, Lagrangian framework with subspace constraints. We track points even through ephemeral occlusions. Experiments demonstrate accuracy superior to the state of the art while tracking more points through more frames.

1. Introduction

Until recently, most work on video motion analysis focused on extracting optical flow fields from consecutive pairs of frames. Borrowing terms from fluid mechanics, a set of motions computed from a sequence of successive frame pairs can be seen as an Eulerian representation of overall flow. In contrast, the Lagrangian approach associates each world point explicitly with its entire image trajectory over time. These extended trajectories provide crucial information for video segmentation and analysis.

In this paper, we show how to compute dense Lagrangian trajectories over extended sequences and in the presence of significant occlusions. We also reconstruct the motion of points that undergo ephemeral occlusions that occur when, say, a car drives behind a telephone pole or a walking person's torso is partially occluded by a swinging arm.

In principle, integration over time can convert an Eulerian representation (optical flow) to a Lagrangian one (trajectories), but this can cause small flow errors to integrate into large trajectory drifts. Also, occlusions are hard to detect from pairs of frames, and computed trajectories may then jump undetected from occluded to occluding object.

In contrast, our Lagrangian approach yields long-range motion *and* allows more reliable detection of occlusions. Evidence from many frames better distinguishes brightness changes caused by occlusion from those due to noise. Also, long-term constraints can be imposed on trajectories to both regularize the solution and help bridge trajectories through ephemeral occlusions. Specifically, we assume — as is often done in recent literature — that trajectories are close to a low-dimensional subspace for which we estimate a basis. We then employ variational methods to solve for the best-fit coefficients of the motion trajectories in this basis.

A point can become occluded only when another point occludes it. Consequently, explicit knowledge of trajectories allows casting occlusion detection as the choice of which of several competing trajectories is chosen to "own" a given pixel in the video. Ownership labels are assigned as part of our optimization and in turn label points along each trajectory as either visible or invisible. If occlusions are of short duration, even the invisible parts of a trajectory can be estimated so that a point that disappears and reappears can be recognized to be the same point.

In summary, occlusion detection and multi-frame motion estimation are strongly coupled problems and addressing one helps solve the other in fundamental ways. Our experiments show the benefits of our approach, which include dense point tracking, interpolation of motion through occlusions, and detailed motion segmentation.

2. Related work

Most optical flow papers since Horn and Schunck [12] have used variational methods to minimize an energy functional that rewards fit to data and a smooth field, with robust norms replacing quadratic ones in modern approaches (see Sun *et al*. [21] for an overview of current best practices).

Focusing on the most closely related approaches for brevity, Sundaram *et al*. [23] integrate the optical flow fields computed by Brox and Malik [7] over time to convert from an Eulerian representation to Lagrangian trajectories. Both their motion estimation and their occlusion detection are based on consecutive frame pairs, and time integration is *post facto*. Sand and Teller [19] add a refinement step on the computed trajectories, and terminate these based, again, on local information. Because of their Eulerian approach, both Sundaram *et al*. and Sand and Teller only track points that are never occluded — a rare occurrence in general video.

Expansion of point trajectories into a low-dimensional basis underlies most recent work on multi-frame motion estimation. Early methods based on rank constraints from single [24] or multiple [9] rigid motions or even on non-rigid motion [5] assume that point trajectories are precomputed. More recent methods use rank constraints to solve directly for optical flow in rigid scenes [14], or sample trajectory space to solve for non-rigid motion [25], but do so sparsely in space. Garg *et al*. [10] use subspace constraints, and Volz *et al.* [26] use parametric methods for very short, 5-frame sequences. Some optical flow methods parameterize individual flow fields using basis flow fields that are either pre-defined [17] or learned from a training set of flow fields [4]. These optical flow methods parameterize motion in space, rather than in time; the parametric form is used to make the spatial smoothness assumption more realistic not to leverage temporal consistency to improve trajectory accuracy. All these methods ignore occlusions.

If addressed at all, occlusions are either handled indirectly through robust metrics [7, 15, 26] or, when modeled explicitly [2, 13, 20, 21, 28, 29], they are detected from spatially and temporally local information. One exception [22] encourages space and time smoothness in a layer-labeling framework, but without enforcing temporal consistency. Another is the work by Apostoloff and Fitzgibbon [1], who detect occlusions by classifying larger space-time patches. But, they do not estimate motion and assume locally linear, primarily horizontal flow.

In contrast, we combine robust (non-parametric) spatial regularization with explicit reasoning about occlusions in a variational, Lagrangian framework with subspace constraints to solve for trajectories directly from video data. We learn new basis trajectories for each sequence and reason globally about occlusions.

3. Variational trajectories with occlusions

Let $\mathbf{p}$ be the location of a point in a reference frame of a video clip. To introduce our technique, we assume that there is one such frame selected and that it occurs at $t = 0$. We will relax this restriction later. We assume that the trajectory the point follows through the sequence can be described in terms of a small number of basis trajectories $\boldsymbol{\varphi}_k(t)$:

$$\mathbf{x}(\mathbf{p}, t) = \mathbf{p} + \sum_{k=1}^{K} c_k(\mathbf{p})\boldsymbol{\varphi}_k(t) \quad \text{with} \quad \boldsymbol{\varphi}_k(0) = 0\,. \quad (1)$$

Consistently with the Lagrangian view, we abbreviate notation for the image intensity function $I(\mathbf{x}(\mathbf{p}, t), t)$ along the trajectory originating at point $\mathbf{p}$ to $I(\mathbf{p}, t)$. If the image intensity along a trajectory is constant, $I(\mathbf{p}, t) = I(\mathbf{p}, 0)$ whenever the point is visible. A visibility flag $\nu(\mathbf{p}, t) \in \{0, 1\}$ indicates if the point is occluded (0) or visible (1) at time t. We cast solving for the unknown coefficients $\mathbf{c} = (c_1(\mathbf{p}), \ldots, c_K(\mathbf{p}))$ in equation (1) and the visibility flag $\nu(\mathbf{p}, t)$ as the following problem:

$$\arg\min_{\mathbf{c},\nu} \left(\lambda_1 E_D(\mathbf{c}, \nu) + E_S(\mathbf{c}) + \lambda_2 E_V(\nu)\right)\,. \quad (2)$$

The data term E_D penalizes intensity changes in visible intervals using the robust function $\Psi(s) = \sqrt{s^2 + \epsilon^2}$:

$$E_D(\mathbf{c}, \nu) = \int_\Omega \int_T \nu(\mathbf{p}, t)\, \Psi\left(I(\mathbf{p}, t) - I(\mathbf{p}, 0)\right)\, d\mathbf{p}\, dt\,. \quad (3)$$

The total-variation smoothness term

$$E_S(\mathbf{c}) = \int_\Omega \sum_k g(\mathbf{p})\, \|\nabla_2 c_k(\mathbf{p})\|_1\, d\mathbf{p} \quad (4)$$

where $g(\mathbf{p}) = \left(\|\nabla_2 I(\mathbf{p}, 0)\|_2 + 1\right)^{-1}$ adds weighted, inhomogeneous spatial regularization. The symbol ∇_k is the gradient in the first k coordinates.

The third term, $E_V(\nu)$, imposes soft constraints on the visibility field ν through two penalties,

$$\begin{aligned} E_V(\nu) \quad &= \quad \underbrace{\int_\Omega \int_T (\nu(\mathbf{p}, t) - \nu^{(0)}(\mathbf{p}, t))^2\, d\mathbf{p}\, dt}_{E_{DV}(\nu)} \\ &+ \quad \underbrace{\int_\Omega \int_T \|W \nabla_3 \nu(\mathbf{p}, t)\|_1\, d\mathbf{p}\, dt}_{E_{SV}(\nu)}\,. \end{aligned} \quad (5)$$

Section 4.3 shows how to compute a per-trajectory visibility estimate $\nu^{(0)}$. $E_{DV}(\nu)$ anchors the global estimate ν to $\nu^{(0)}$. $E_{SV}(\nu)$ encourages visibility to be smooth in space and time. The matrix $W = \mathrm{diag}([w_1\ w_2\ w_3])$ levies different penalties for different trajectories and times. Because neighboring trajectories in the reference frame may no longer be close spatial neighbors at a later time, the two spatial penalties decrease as distance between initial neighbors increases. The temporal weight w_3 aligns visibility changes with brightness changes. (See Section 4.3.)

Connection to two-frame optical flow. Our formulation subsumes two-frame methods by letting $\boldsymbol{\varphi}_1(t) = (t, 0)$ and $\boldsymbol{\varphi}_2(t) = (0, t)$. The two frames are at times $t = 0$ and $t = 1$, and the two flow components are $u = c_1$ and $v = c_2$.

Finding a trajectory basis. The basis functions $\boldsymbol{\varphi}_k(t)$ can be either given or inferred from the data. If a sparse set of points can be tracked using a frame-to-frame tracker throughout the sequence, a basis can be found by PCA. Otherwise, the problem of finding a basis becomes one of low-rank matrix completion [8]. In this work, we tracked points visible throughout the video with standard methods [16] and automatically selected K by adding principal components until the basis represents the sparse tracks with a maximum error of two pixels. A rigid object requires at most $K = 4$

basis trajectories under orthographic or weak-perspective projection [24]. Similarly, a few basis trajectories often suffice to represent the motion of non-rigid objects over many frames [5]. Thus, K is often small in practice.

Using multiple reference frames. The Lagrangian formulation requires that each trajectory be associated with a unique identifier. Using the position in the first frame of the sequence provides such an identifier for any point visible in the first frame, but not for points that are not visible. We therefore use additional reference frames to identify initially occluded points. In our experiments, we use the first and last frames for reference. We solve for motion independently for each reference frame but couple them during occlusion estimation as described in Section 4.3.

4. Solution method

To approximately solve (2) from an initial solution $\mathbf{c}^{(0)}, \nu^{(0)}$ we alternate between solving for trajectory coefficients $\mathbf{c}$ with visibility ν fixed and solving for visibility with trajectories fixed. The next three sections describe the optimization method, how we find $\mathbf{c}^{(0)}$, and how we compute both $\nu^{(0)}$ for initialization and ν during optimization.

4.1. Minimization

We approximately solve the minimization problem (2) by alternating between solving for trajectories with visibility fixed and solving for visibility with trajectories fixed:

$$\arg\min_{\mathbf{c}} \left(\lambda_1 E_D(\mathbf{c}, \nu) + E_S(\mathbf{c})\right) \tag{6}$$

$$\arg\min_{\nu} \left(\lambda_1 E_D(\mathbf{c}, \nu) + \lambda_2 E_V(\nu)\right) \ . \tag{7}$$

The terms E_D and the penalty E_{DV} in E_V are differentiable in the unknowns, while E_S and E_{SV} are sums of L_1 terms. Thus, both problems can be written in the form

$$\arg\min_{\mathbf{u}} \|\mathbf{F}(\mathbf{u})\|_1 + H(\mathbf{u}) \tag{8}$$

where both the components of $\mathbf{F}$ and the scalar function H are convex, and H is differentiable. The L_1 and L_2 terms in problems of this form can be separated [27] by introducing an auxiliary vector $\mathbf{d} = \mathbf{F}(\mathbf{u})$ so that the problem becomes

$$\arg\min_{\mathbf{u},\,\mathbf{d}} \|\mathbf{d}\|_1 + H(\mathbf{u}) \quad \text{such that} \quad \mathbf{d} = \mathbf{F}(\mathbf{u}) \ . \tag{9}$$

The so-called *Split Bregman* method further defines the discrepancy $\mathbf{b} = \mathbf{d} - \mathbf{F}(\mathbf{u})$ and solves problem (9) by the efficient and numerically stable Algorithm 1, where

$$\text{shrink}(x, \gamma) = \frac{x}{|x|} \max(|x| - \gamma,\, 0) \ . \tag{10}$$

The only expensive computation is the first line in the **while** loop; in both problems (6) and (7) this leads to the minimization of a differentiable function. Specifically, we solve the non-linear Euler-Lagrange equations for problem (6) by nested fixed point iterations [6]. The matrix of the resulting sparse, symmetric linear system is $NMK \times NMK$ for $M \times N$ image frames and K trajectory basis functions, and we solve that by Gauss-Seidel iterations.

Algorithm 1 The Split Bregman algorithm [11]. The shrink operator is defined in equation (10).

Input: Start point $\mathbf{u}^{(0)} \in \mathbb{R}^m$ and tolerance $\epsilon > 0$
  $\mathbf{b}^{(0)} = 0\,,\ \mathbf{d}^{(0)} = 0$
  while $k = 0$ or $\|\mathbf{u}^{(k)} - \mathbf{u}^{(k-1)}\| > \epsilon$ **do**
    $\mathbf{u}^{(k+1)} = \arg\min_{\mathbf{u}} H(\mathbf{u}) + \lambda \|\mathbf{d}^{(k)} - \mathbf{F}(\mathbf{u}) - \mathbf{b}^{(k)}\|_2^2$
    for $j = 1 \to m$ **do**
      $d_j^{(k+1)} = \text{shrink}\left(F_j(\mathbf{u}) + b_j^{(k)}, 1/(2\lambda)\right)$
    end for
    $\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} + \mathbf{F}(\mathbf{u}^{(k+1)}) - \mathbf{d}^{(k+1)}$
  end while

During these iterations, we enhance non-local smoothness by replacing the trajectory coefficients with the results of a weighted median filter in the outer fixed point iterations as suggested by Sun *et al.* [21]. We use a 15×15 local neighborhood and weight pixels with a Gaussian function of spatial distance, deviation from intensity in the reference frame, and distance between visibility states.

We solve problem (7) by relaxing the domain for ν from $\{0, 1\}$ to $[0, 1]$ and round the solution at $1/2$. We solve the Euler-Lagrange system of linear equations for the relaxed version by Gauss-Seidel.

4.2. Trajectory initialization

We use sparse tracks from a KLT tracker to compute an initial value $\mathbf{c}^{(0)}$ for the coefficients $\mathbf{c}$ in equation (1). The sparse tracks define a set of candidate motions that may be used. At each pixel in a reference frame, we score candidate motions with a cost that measures variations of intensity in a tube along each motion and within a few frames from the reference frame. We use tubes at a variety of scales and select the motion with the lowest intensity variation at the finest scale at which we can distinguish candidates, assigning coefficients according to this labeling. Regions that are occluded within a few frames will tend to have very large costs for the selected motion because brightness constancy is violated at occlusions. We mark regions with lowest variation greater than two standard deviations above the mean as unreliable. Finally, we transform per-pixel motion estimates into piecewise-constant regions by applying a weighted median filter. Weights are zero for unreliable pixels. Recent work [18] on extracting regions from point trajectories may provide a more principled initialization.

4.3. Visibility estimation

Visibility estimation is composed of two steps: (1) finding an initial per-trajectory estimate $\nu^{(0)}$ and (2) smoothing the local estimate across trajectories using equation (7) to produce a global estimate ν.

Local estimate. Let $\mathbf{p}_r$ identify a trajectory from reference frame r. We measure violations of brightness constancy for this trajectory at time t by

$$D(\mathbf{p}_r, t) = \Delta(\mathbf{p}_r, t, t) + \Delta(\mathbf{p}_r, t, r) \tag{11}$$

where

$$\begin{aligned}\Delta(\mathbf{p}_r, t, t') &= \iiint_{-\infty}^{\infty} w(\boldsymbol{\xi}, \tau - t) \; |I(\mathbf{x}(\mathbf{p}_r, \tau) + \boldsymbol{\xi}, \tau) \\ &- I(\mathbf{x}(\mathbf{p}_r, t') + \boldsymbol{\xi}, t')| \; d\boldsymbol{\xi}\, d\tau \end{aligned} \tag{12}$$

and $w(\boldsymbol{\xi}, \tau) = \exp[-(\|\boldsymbol{\xi}\|^2 + \tau^2)/\sigma^2]/\sigma$. The value $\Delta(\mathbf{p}_r, t, t)$ measures Gaussian-weighted, absolute image differences between the current image patch, centered at $\mathbf{x}(\mathbf{p}_r, t)$ and time t, and the volume of image values obtained by transporting the current patch through the motion $\mathbf{x}(\mathbf{p}_r, t)$ over a small time interval centered at t. The value $\Delta(\mathbf{p}_r, t, r)$ is similar, but compares the volume to the *reference* patch, centered at $\mathbf{p}_r = \mathbf{x}(\mathbf{p}_r, r)$ and time r. We found including both terms improved performance.

If the point $\mathbf{p}_r$ is visible at time t, we expect $D(\mathbf{p}_r, t)$ to be low and high otherwise. Crucially, if $\mathbf{p}_r$ is occluded, there must be some *other* trajectory $\mathbf{q}$ that is visible at time t and hides $\mathbf{p}_r$; we expect $D(\mathbf{q}, t) < D(\mathbf{p}_r, t)$. Here, trajectories based in different reference frames are coupled; trajectories in one reference frame may be the occluders of trajectories in a different reference frame. Let $P(\mathbf{x}, t)$ be the set of all trajectories (from any reference frame) passing within a fixed neighborhood of a pixel $(\mathbf{x}, t)$, and let

$$\mathbf{p}^*_{\mathbf{r}^*}(\mathbf{x}, t) = \arg \min_{\mathbf{p}_r \in P(\mathbf{x}, t)} D(\mathbf{p}_r, t) \tag{13}$$

denote the best-fitting, *controlling trajectory* at this pixel.

The sets $P(\mathbf{x}, t)$ for all points in the video can be computed in linear time with a single pass along each trajectory. On a second pass along each trajectory, we compare the reference location of the current trajectory, $\mathbf{p}_r$, to the location of the controlling trajectory, $\mathbf{x}(\mathbf{p}^*_{r^*}, r)$. When the two locations are distant, the trajectory $\mathbf{p}_r$ is likely occluded by $\mathbf{p}^*_{r^*}$ at time t. In our implementation, we make the visibility decision more robust by declaring an occlusion of trajectory $\mathbf{p}_r$ at time t when the median distance of the k best-fitting trajectories at $\mathbf{x}(\mathbf{p}_r, t)$ is more than a few pixels.

The technique above overestimates occlusions at shear boundaries because it accumulates intensity differences across the discontinuity. Let $D_0(\mathbf{p}_r, t)$ denote $D(\mathbf{p}_r, t)$ without spatial integration. Visibility estimation improves if we set $\nu^{(0)}(\mathbf{p}_r, t)$ to 1 when $D_0(\mathbf{p}_r, t) < D(\mathbf{p}^*_{r^*}, t)$.

Our procedure for visibility estimation meshes intimately with Lagrangian motion estimation. Accumulating evidence over multiple frames makes it easier to distinguish intensity changes due to noise from those due to occlusions. Furthermore, occlusions are only declared when an occluder, $\mathbf{p}^*_{r^*}$, can be found; this avoids the trivial minimizer of equation (2). In practice, the distance between the occluder $\mathbf{x}(\mathbf{p}^*_{r^*}, r)$ and the occluded $\mathbf{p}_r$ increases rapidly with t, simplifying the visibility decision.

Global estimate. With $\nu^{(0)}$ computed, we can solve problem (7). The weights in E_{SV} of equation (5) are $w_i = \left(\left\| \frac{\partial \mathbf{x}(\mathbf{p}, t)}{\partial p_i} \right\|^2 + 1 \right)^{-1/2}$ for $i = 1, 2$ and $\mathbf{p} = (p_1, p_2)$ and $w_3 = \left[\left(\frac{\partial^2 D_0}{\partial t^2} \right)^2 + 1 \right]^{-1/2}$. The two spatial weights decrease the penalty for spatial discontinuities if trajectories have diverged by frame t. The remaining weight encourages jumps in $\nu(\mathbf{p}, t)$ to align with brightness changes (inflection points of D_0) along a trajectory. The solution to problem (7) will have real values for ν; we round at $1/2$ to recover binary values.

5. Experiments

Although valuable for two-frame optical flow, the Middlebury benchmark [3] is not well-suited to evaluating long-range motion estimation. Our approach interpolates trajectories through temporary occlusions, but the eight frames of the Middlebury sequences do not allow enough time for occluded regions to reappear. Also, the benchmark does not give ground truth correspondences over the full eight frames. For benchmarks with longer sequences, ground truth is available either only for points that are never occluded [23] or only in the first and last frame [19].

To demonstrate our ability to maintain dense correspondences across temporary occlusions, we validate our work on three sequences (see Table 1) for which we provide two types of ground truth: either spatially sparse and temporally dense — obtained by manually tracking a few points throughout the video — or temporally sparse and spatially dense — obtained by hand-segmenting every tenth frame, plus the first and the last. For hand-tracked points, we also provide a true visibility flag $\nu^*(\mathbf{p}, t)$ in every frame. We compare our results to those by Sundaram *et al.* [23] ('LDOF' below) using code provided by the authors.

In all experiments, we use $\lambda_1 = \lambda_2 = 1$ with pixel intensities in $[0, 255]$. We run 20 iterations alternating between solving for motion with visibility fixed and solving for visibility with motion fixed and complete 30 Bregman iterations within each component optimization. We observed small variations in the results if these parameters are changed within reason.

Sequence	Size	K	Avg. disp.	Max disp.	# LDOF trajectories	# Full LDOF
Checkerboards	$100 \times 100 \times 15$	1	3.1	28.0	1962 (19.6%)	1106 (11.1%)
Flowerbed	$120 \times 175 \times 29$	2	33.7	78.6	10894 (51.9%)	5959 (28.4%)
Marple	$175 \times 225 \times 25$	5	101.9	140.8	22440 (57.0%)	5050 (12.8%)

Table 1: Details on the three test sequences. K is the number of basis trajectories automatically selected by our algorithm and indicates complexity of motion. Average and maximum displacement (disp.) measure the amount of motion between the first and last frames of the sequence. Values are distances in pixels. Remaining columns report the number of pixels in the first frame with an LDOF trajectory of any length, or a full LDOF trajectory.

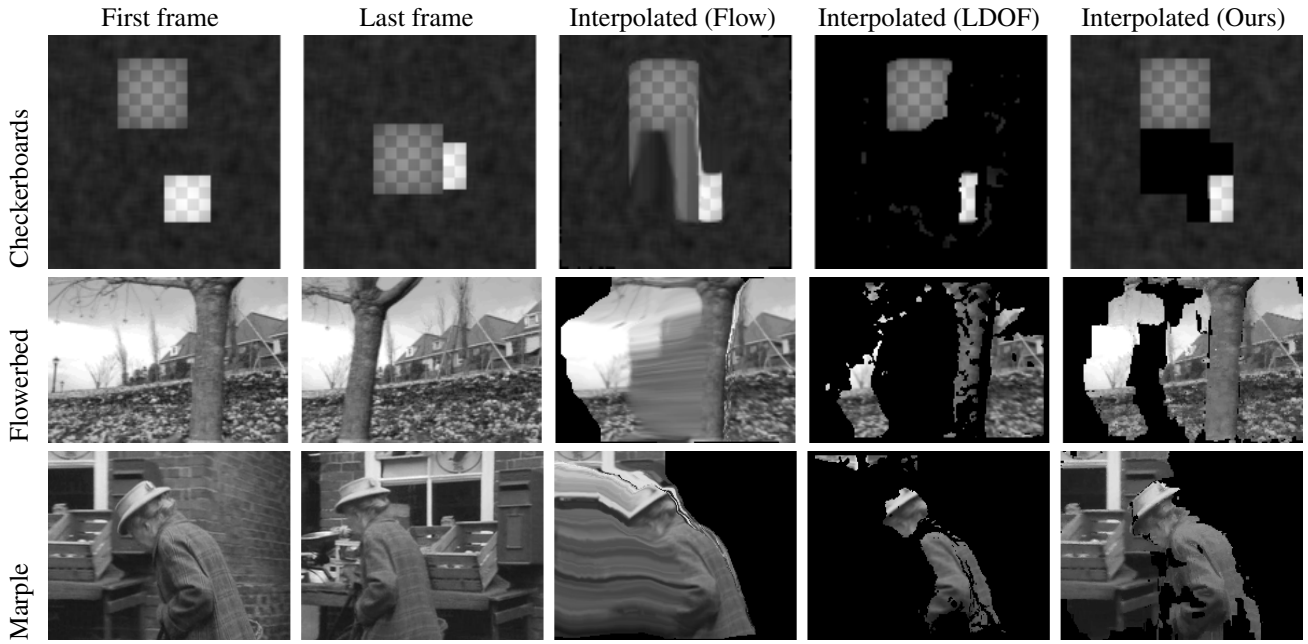

Figure 1: Warping last frame of sequence back to first frame using correspondences between the first and last frame. Last three columns show interpolation results based on integrated optical flow (middle), LDOF (middle-right), and our method (right). Black regions either lack correspondences (Flow, LDOF) or are determined to be occluded (Ours). Our algorithm maintains correspondence across temporary occlusions and so can correctly interpolate previously occluded background regions. The correct answer matches the image in column 1, but with points not visible in column 2 painted black.

5.1. Performance metrics

To evaluate the accuracy of estimated locations and occlusions, we compute dense error metrics over all trajectories. We also investigate the accuracy of both terms in detail using the sparse, hand-tracked, ground truth trajectories.

Dense metrics. It is prohibitively expensive to label correspondences for every point in our image sequences. Instead, we measure dense positional accuracy of our estimated trajectories using normalized interpolation error. Correspondences provided by trajectories allow us to warp any frame to a reference frame. The last column of Figure 1 shows the interpolated images when the last frame is warped to the first frame for each sequence. Black pixels in the interpolated frame are those flagged as occluded in the final frame and are discounted when computing intensity differences.

We compare the performance of our algorithm to two alternative methods. The first is a naive conversion from an Eulerian framework to a Lagrangian framework (labeled 'Flow' in figures and tables), where correspondences between frames are computed by simply integrating and interpolating successive flow fields without any occlusion reasoning. We use the LDOF flow fields [7] – the same flow fields used in the LDOF tracker. This technique provides correspondences for every pixel in the first frame, as long as its interpolated trajectory remains in the field of view. However, these correspondences are wildly inaccurate for temporarily occluded regions. The second competing method is the full LDOF tracker [23], which adds occlusion reasoning to the naive alternative. The tracker returns correspondences for most but not all points in the first frame, but terminates trajectories at detected occlusions and near motion

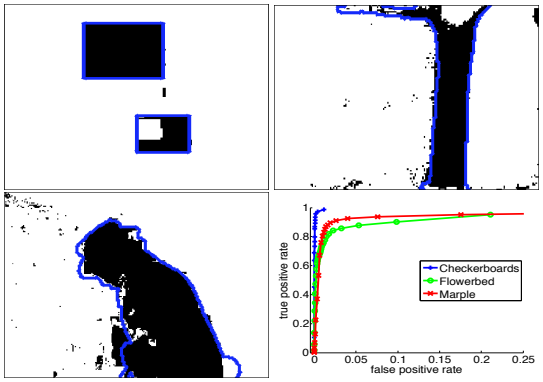

Figure 2: Detected occluders for each sequence at consensus best threshold (2 frames) and ROC curves varying threshold for each sequence. Black pixels are detected occluder regions; blue contours outline ground truth occluder regions. Results shown correspond to overall precision of 0.92 and recall of 0.90. *View in color.*

boundaries. When interpolating, we ignore pixels with no correspondence reported. Quantitative results for all three methods are given in Panel 1 of Table 2.

We measure our occlusion accuracy by classifying trajectories as *occluders* or *non-occluders*. Because our algorithm provides interpolated positions for occluded trajectories during the occlusion, we can mark the regions in the video when these occlusions occur. Setting a threshold on the number of frames during which a visible trajectory ($\nu(\mathbf{p},t)=1$) passes through a marked region classifies points as occluders. We use the ground truth segmentation of the first frame to measure our performance in terms of precision and recall. Alternative methods do not provide the information needed to extract occluders in a similar fashion, so we do not report results for those techniques on this metric. Figure 2 shows example occluder detections for each sequence, as well as ROC curves showing the tradeoff between true and false positives for the sequences.

Metrics based on spatially sparse ground truth. For a more detailed investigation of the positional accuracy of our trajectories and the temporal fidelity of our occlusion estimate, we use a sparse set of hand-tracked ground truth trajectories, approximately uniformly spaced in the first frame of the sequence. We measure the mean and maximum distance between our computed and ground truth trajectories whenever $\nu^*(\mathbf{p},t)=\nu(\mathbf{p},t)=1$ (visible). We measure performance on two sets of tracks: all labeled tracks also tracked for at least one frame by the LDOF tracker, and the set of labeled tracks for which the LDOF tracker returns a full trajectory. Our algorithm always returns a full trajectory. We compare our results to trajectories returned by the naive Flow method and the full LDOF tracker. For competing algorithms, we measure distance between ground truth

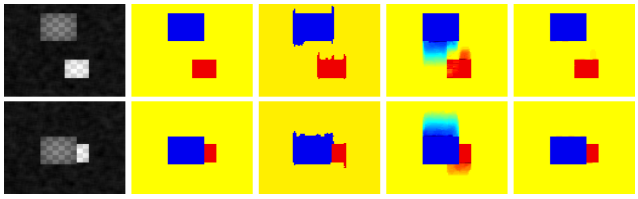

Figure 3: Explicit handling of occlusions is crucial. Leftmost column: first (top) and last (bottom) frame of the Checkerboards sequence. Remaining four columns show color-coding of coefficients for the single basis trajectory at each pixel, using either the first or last frame as reference. From left to right: true coefficients, initialization, coefficients without explicit occlusions, and coefficients with occlusions. Ignoring occlusions results in large motion errors in occluded regions. *View in color.*

and estimated trajectories in any frame where $\nu^*(\mathbf{p},t)=1$ and the algorithm returns correspondences. This is a smaller set for the LDOF tracker than for either our method or the Flow method.

We measure occlusion detection performance using precision and recall statistics that compare true ($\nu^*(\mathbf{p},t)$) and computed ($\nu(\mathbf{p},t)$) visibility values. This is essentially the reverse of our dense occluder detection metric. For our method, an occlusion is a change from $\nu(\mathbf{p},t)=1$ to $\nu(\mathbf{p},t)=0$ and a disocclusion is the reverse. For the LDOF tracker, we consider a terminated LDOF trajectory to be an occlusion detection. LDOF cannot detect disocclusions because it considers different trajectory segments for the same point to be trajectories for distinct points. The naive Flow method does not detect occlusions. Results for the positional accuracy and occlusion detection of the competing algorithms are shown in Panels 2 and 3 of Table 2.

5.2. Remarks

We now report observations and details about our experiments with each sequence. See Table 1 for details on the sequences and the density of trajectories returned by LDOF.

Checkerboards. We include a synthetic sequence to illustrate the necessity of explicit occlusion detection. We estimate trajectories twice, with the same parameters and initialization each time. The first time, we estimate occlusions using the procedure in Section 4.3. The second time, we set $\nu(\mathbf{p},t)=1$ for all $\mathbf{p}$ and t, relying solely on the robust data term to discount intensity changes due to occlusion. Figure 3 shows that trajectories in the occluded region are recovered accurately when occlusions are handled, but exhibit significant errors when they are not.

Our algorithm with occlusion handling outperforms competing methods in every metric. Like our method, the Flow method returns a trajectory for every point in the

		Panel 1	Panel 2				Panel 3		
Sequence	Method	NE	Mean (maximum) position error				Occlusion detection		
			all labeled		full only		Precision	Recall	F-meas.
Checkerboards	Ours	0.306	0.001	(0.05)	0.001	(0.02)	1.000	0.951	0.975
	LDOF	0.337	0.567	(23.3)	0.208	(13.6)	0.500	0.839	0.626
	Flow	3.247	1.424	(26.3)	0.208	(13.5)	—	—	—
Flowerbed	Ours	1.583	0.644	(4.6)	0.725	(4.6)	0.980	0.877	0.926
	LDOF	1.476	0.760	(5.6)	0.845	(5.6)	0.634	0.959	0.763
	Flow	6.427	4.013	(50.5)	0.848	(5.7)	—	—	—
Marple	Ours	1.645	1.295	(7.0)	1.672	(6.3)	0.982	0.935	0.958
	LDOF	2.106	1.331	(6.4)	1.801	(6.4)	0.535	1.000	0.697
	Flow	13.511	15.933	(157.7)	1.804	(6.5)	—	—	—

Table 2: Performance comparison between our method ("Ours"), the Large-Displacement Optical Flow tracking method [23] ("LDOF"), and simple temporal integration of LDOF [7] flow fields ("Flow"). Better values are highlighted in yellow.
Panel 1: Normalized interpolation error (NE) measures dense trajectory accuracy. Differences are measured between the first frame and successive frames warped back to the first frame. Our method maintains accuracy while returning more correspondences across more frames (note the decrease in the amount of black in images in the last column of Figure 1). The large values for the Flow method are the result of undetected occlusions.
Panel 2: Positional accuracy is measured by the mean and maximum error in pixels between computed and ground truth trajectories. We measure error on hand-labeled trajectories that are tracked by LDOF for at least one frame (all labeled), and on the subset of those tracks for which LDOF returns a complete trajectory (full only).
Panel 3: Occlusion detection performance measures precision and recall for occlusions. Our method uses a fixed threshold of $1/2$ on $\nu(\mathbf{p}, t)$. LDOF track terminations are considered to be occlusion detections. LDOF cannot detect disocclusions, while we do by tracking through occlusions. We achieve competitive recall and superior precision, resulting in increased F-measure. The Flow method does not detect occlusions.

first frame, but its accuracy suffers from lack of occlusion modeling. As Table 2 shows, the performance of the Flow method is unacceptable on this and the other two sequences. LDOF performs surprisingly poorly on this sequence because its estimated optical flow fields are oversmooth. Missing motion discontinuities result in tracks on the background dragging along with the foreground and accumulating large positional error by the end of the sequence. Our algorithm returns much more accurate motion discontinuities, resulting in much better positional accuracy.

Our only error on this sequence is a row of missed occlusions between the first and second frame at the occluding edge of each checkerboard. By the second frame, not enough motion has accumulated for the distance between the occluded and occluding trajectory in the reference frame to be significant. The LDOF method aggressively terminates trajectories near shear motion discontinuities (vertical edges of the checkerboards), resulting in a large number of false positives and a poor precision score. At the same time, recall suffers because many points near the occluding edge are incorrectly assigned to the foreground motion.

Flowerbed. In this well-known sequence, a foreground tree sweeps over a large portion of the background. We maintain tracks across these transient occlusions and accurately identify occlusions and disocclusions. We outperform the LDOF tracker on every sparse metric, both for occlusion detection and positional accuracy. We have slightly higher normalized interpolation error because we interpolate difficult disoccluded regions that the LDOF tracker ignores. When restricted to pixels in common, our normalized interpolation error is 1.19 (compared to 1.476 for LDOF). The majority of our mistakes are occlusions that are detected one frame too late or disocclusions that are detected one frame too early. Our recall performance is still good in spite of these errors. Furthermore, our ability to detect disocclusions vastly improves our precision numbers compared to LDOF, resulting in overall better performance on the task. Our causal occlusion reasoning allows us to accurately segment the tree from the background (Figure 2).

Marple. This sequence contains challenging non-rigid motion and occlusions due to the motion of the arm and body in the foreground. The woman occludes the majority of the background in at least one frame. As a result, full trajectories from LDOF are restricted almost entirely to the interior of the woman's body. With non-rigid motions, we lack theoretical guarantees that the low-rank assumption is valid. With its non-parametric representation, LDOF would be able to represent motions of any rank exactly. Nonetheless, a few basis trajectories suffice to reconstruct the motion in this sequence, and we outperform LDOF on all metrics ex-

cept the maximum positional error. LDOF again achieves better recall but much poorer precision resulting in worse occlusion detection performance overall. We detect occlusions and disocclusions accurately to within a single frame.

Our results on this sequence demonstrate the benefits of using all frames in the sequence with the Lagrangian approach. The LDOF method constructs trajectories from optical flow fields generated from successive pairs of frames, without any temporally global reasoning. As a result, its trajectories are susceptible to drift and errors caused by image noise or motion blur. By integrating intensity information along the entire trajectory through all frames, our results are less affected by these transient sources of error.

6. Discussion

Our approach to motion estimation models trajectories in a subspace and handles pixel visibility explicitly. We recover fully dense, accurate trajectories across ephemeral occlusions and leverage global information for visibility estimation. Our experiments demonstrate that explicitly handling occlusions is crucial for trajectory accuracy and that our method is superior to the present state of the art.

We currently compute a motion basis from points that are visible throughout the video. We are working to build useful motion bases from incomplete trajectories. We are also exploring ways to select reference frames automatically and enforce consistency across reference frames. We also plan to improve efficiency. It currently takes about 20 minutes to solve for the 15-frame trajectories of the Checkerboards sequence. Mapping on the GPU will reduce computation time dramatically, as many steps of our algorithm can be performed in parallel.

Acknowledgment. The authors thank the anonymous reviewers for their comments. This work is supported by the National Science Foundation under Grant No. IIS-1017017.

References

[1] N. Apostoloff and A. Fitzgibbon. Learning spatiotemporal T-junctions for occlusion detection. *CVPR*, 2005. 2

[2] A. Ayvaci, M. Raptis, and S. Soatto. Occlusion detection and motion estimation with convex optimization. *NIPS*, 2010. 2

[3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *ICCV*, 2007. 4

[4] M. Black, Y. Yacoob, A. Jepson, and D. Fleet. Learning parameterized models of image motion. *CVPR*, 1997. 2

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. *CVPR*, 2000. 2, 3

[6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *ECCV*, 2004. 3

[7] T. Brox and J. Malik. Large displacement optical flow. *CVPR*, 2009. 1, 2, 5, 7

[8] E. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010. 2

[9] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179, 1998. 2

[10] R. Garg, L. Pizarro, D. Rueckert, and L. Agapito. Dense multi-frame optic flow for non-rigid objects using subspace constraints. *ACCV*, 2010. 2

[11] T. Goldstein and S. Osher. The Split Bregman Method for L1-Regularized Problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. 3

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

[13] S. Ince and J. Konrad. Occlusion-aware optical flow estimation. *IEEE Trans. on Image Processing*, 17(8):1443–1451, 2008. 2

[14] M. Irani. Multi-frame correspondence estimation using subspace constraints. *IJCV*, 48(3):173–194, 2002. 2

[15] K. Jia, X. Wang, and X. Tang. Optical flow estimation using learned sparse model. *ICCV*, 2011. 2

[16] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, 1981. 2

[17] T. Nir, A. Bruckstein, and R. Kimmel. Over-parameterized variational optical flow. *IJCV*, 76(2):205–216, 2008. 2

[18] P. Ochs and T. Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. *ICCV*, 2011. 3

[19] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 80(1):72–91, 2008. 1, 4

[20] C. Strecha, R. Fransens, and L. Van Gool. A probabilistic approach to large displacement optical flow and occlusion detection. *Statistical methods in video processing*, pages 25–45, 2004. 2

[21] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. *CVPR*, 2010. 1, 2, 3

[22] D. Sun, E. Sudderth, and M. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. *NIPS*, 2010. 2

[23] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. *ECCV*, 2010. 1, 4, 5, 7

[24] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, 1992. 2, 3

[25] L. Torresani and C. Bregler. Space-time tracking. *ECCV*, 2002. 2

[26] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. Modeling temporal coherence for optical flow. *ICCV*, 2011. 2

[27] Y. Wang, W. Yin, and Y. Zhang. A fast algorithm for image deblurring with total variation regularization. Technical Report TR07-10, CAAM, Rice University, 2007. 3

[28] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. *ECCV*, 2006. 2

[29] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *CVPR*, 2010. 2