Drift Robust Non-rigid Optical Flow Enhancement for Long Sequences

Wenbin Li, Darren Cosker and Matthew Brown

Department of Computer Science, University College London, WC1E 6BT, UK
E-mail: w.li@cs.ucl.ac.uk
Centre for the Analysis of Motion, Entertainment Research and Applications (CAMERA), University of Bath, BA2 7AZ, UK. E-mail: d.p.cosker@bath.ac.uk
Google, Mountain View, CA 94043, US
E-mail: m.brown@bath.ac.uk

Abstract. It is hard to densely track a nonrigid object in long term, which is a fundamental research issue in the computer vision community. This task often relies on estimating pairwise correspondences between images over time where the error is accumulated and leads to a drift. In this paper, we introduce a novel optimisation framework with an Anchor Patch constraint. It is supposed to significantly reduce overall errors given long sequences containing nonrigidly deformable objects. Our framework can be applied to any dense tracking algorithm, e.g., optical flow. We demonstrate the success of our approach by showing significant error reduction on 6 popular optical flow algorithms applied to a range of real-world nonrigid benchmarks. We also provide quantitative analysis of our approach given synthetic occlusions and image noise.

Keywords: Computer Vision, Dense Tracking, Anchor Patch, Optical Flow, Drift, Long Sequence

1. Introduction

Tracking a set of landmark points through multiple images is a fundamental research issue in computer vision. We define tracking in this work as the estimation of corresponding sets of vertices, pixels or landmark points between a reference frame and any other frame in the same image sequence. In the last two decades, optical flow has become a popular approach for tracking through image sequences in fields in fields 21,20,12,28. Compared with feature matching methods, e.g., 19, optical flow provides subpixel accuracy and dense correspondence between a pair of images. In this work, we focus in particular on improving tracking in image sequences using optical flow, and our contribution applies to this class of algorithm.

One of the main drawbacks of optical flow is drift 6,15. Errors accumulated between frames over time result in movement away from the correct tracking trajectory. Between single image pairs, this problem may not be noticeable. However, accumulation when tracking across long sequences can be particularly problematic. Several authors have previously attempted to reduce optical flow drift in tracking. DeCarlo et al. 7 introduce contour information on a human face to improve tracking stability, while Borshukov et al. 4 employ manual correction. More recently, Bradley et al. 5 proposed an optimisation method constrained by additional tracking information from multiview video sequences. Beeler et al. 2 then introduced the concept of anchor frames for human face tracking. In this approach, the sequence is decomposed into several clips based on anchor images which are visually similar to a reference frame. Their optimisation method shortens the tracking distance from reference frames to the target frame to help alleviate errors. However,
their approach is domain specific (faces), and assumes that the entire face will return to a neutral expression (the anchor) several times throughout the sequence. In general, it is difficult to label anchor frames on general object sequences with large displacement motion e.g. waving cloth, as there is usually significant deformation between the reference frame and the other frames. In addition, repeated patterns are typically not global as observed in a face (return to a neutral expression). Rather, they occur in smaller local regions at intermittent intervals.

In this work, we focus on tracking long video sequences using optical flow algorithms, and specifically concentrate on reducing drift. The general strategy of our approach is to shorten tracking distances for local regions throughout a long sequence. Our proposed framework combines long term feature matching with optical flow estimation. It may be applied to the tracking of general objects with large displacement motion, and results in a significant reduction in drift. We first detect Anchor Frames for a sequence (Sec. 2). This provides an initial set of start points for tracking the sequence. Our main contribution is extending this approach by proposing the concept of Anchor Patches (Sec. 5). These are corresponding points and patches throughout the sequence which are propagated directly from the reference frame. Our framework substantially reduces overall drift on a tracked image sequence, and may be applied to any optical flow algorithm in a straightforward manner. In our evaluation, we apply the proposed optimisation framework on 6 popular optical flow estimation algorithms to illustrate its applicability. We provide analysis of our method for a sequence (Sec. 4).

2. System Overview

Our proposed optimisation framework reduces overall optical flow drift given long image sequences, and provides additional robustness against other issues such as large displacements and occlusions. The major procedure is shown in Table 1. The aim of our Anchor Patch optimisation Framework (APO) is accurately tracking a mesh denoted by \( M_R = (V_R, E_R, F_R) \) from a reference frame \( I_R \) to every other frame \( I_i \) in the sequence. \( M_i = (V_i, E_i, F_i) \) denotes the corresponding mesh on frame \( I_i \). In the following sections, the four major steps are discussed in detail.


The first step is to compute an optical flow field for a sequence (Sec. 3). In our evaluation, we consider application of our APO framework on a number of dense correspondence optical flow or tracking approaches, e.g. Brox et al. [6], Classic+NL [24] and ITV-L1 [26]. Let \( w_{i\rightarrow i+1} \) denote the optical flow field from frame \( I_i \) to frame \( I_{i+1} \). Similarly we have \( w_{i+1\rightarrow i} \), denoting the optical flow field from frame \( I_{i+1} \) to frame \( I_i \) in the backward direction. The optical flow field between frame \( I_i \) and \( I_j \) where \( i < j \) (Forward direction), is denoted by \( w_{i\rightarrow j} \) as \( w_{i\rightarrow j} = \sum_{i<j} w_{i\rightarrow i+1} \). Similarly, the optical flow field between frame \( I_j \) and \( I_i \) where \( i < j \) (Backward direction), is denoted by \( w'_{j\rightarrow i} \) and is calculated as \( w'_{j\rightarrow i} = \sum_{j>i} w'_{j\rightarrow j-1} \).

In order to evaluate the optical flow at a specific pixel \( x = (x, y)^T \), an Error Score \( E(w) \) is proposed here, where \( w = (u, v)^T \) is the optical flow vector at pixel \( x \). The pixel \( x \) in frame \( I_i \) is matched to pixel \( x' = (x', y')^T \) in frame \( I_{i+1} \) where \( x' = x + w \). The Error Score \( E(w) \) is calculated as the weighted Root Mean Square (RMS) error at a \( 3 \times 3 \) pixel area centred on pixel \( x \) and \( x' \).

\[
E(w) = \sqrt{\frac{\alpha_1 d(x, y) + \alpha_2 d_c(x, y) + \alpha_3 d_d(x, y)}{\alpha_1 + \alpha_2 + \alpha_3}}
\]

\[
d_d(x, y) = d(x - 1, y - 1) + d(x + 1, y + 1) + d(x - 1, y + 1) + d(x + 1, y - 1)
\]

\[
d_c(x, y) = d(x - 1, y) + d(x + 1, y) + d(x, y - 1) + d(x, y + 1)
\]

\[
d(x, y) = |I_i(x, y) - I_{i+1}(x + u, y + v)|^2
\] (1)
Input: A reference frame, a triangle mesh and an image sequence

**Step 1.** Computing Optical flow fields (Sec. 3)
1.1 Compute optical flow fields in both forward \( w_{i \rightarrow i+1} \) and backward \( w'_{i+1 \rightarrow i} \)
1.2 Define the Error Score function

**Step 2.** Detect anchor frames and propagate the entire mesh to these frames (Sec. 4)
2.1 Match SIFT features from the reference to every other frame
2.2 Compute the general Error Score on matchings
2.3 Label the anchor frames from any frame with the low general Error Score

**Step 3.** Label anchor patches on non-anchor frames (Sec. 5)
3.1 Reuse the SIFT feature matching from 2.1
3.2 Propagate patches from the reference using Barycentric Coordinate Mapping

**Step 4.** Track remaining patches from anchor frames to non-anchor frames (Sec. 6)
4.1 Propagate patches from the reference to anchor frames (Sec. 6.1)
   4.1.1 Compute concatenating optical flow field \( w_{R \rightarrow A} \)
   4.1.2 Propagate patches from anchor frames to non-anchor frames using \( w_{R \rightarrow A} \)
   4.1.3 Refine the patches using Error Score
4.2 Propagate patches from anchor frames to non-anchor frames (Sec. 6.2)
   4.2.1 Track the patches from the reference to frame \( i \) using \( w_{A \rightarrow i} \)
   4.2.2 Track the patches from the Nearest Anchor Patches to frame \( i \)
   4.2.3 Eliminate the vertex position conflicts between 4.2.1 and 4.2.2

Output: A mesh tracked throughout the entire image sequence

**Table 1**
The major steps of the Anchor Patch optimisation framework.

---

Fig. 2. **Step One.** The optical flow fields are computed in both forward \( w_{i \rightarrow i+1} \) and backward \( w'_{i+1 \rightarrow i} \) directions between every adjacent images pair in the sequence where the first frame is labelled as a reference frame.

Where \( \alpha_1, \alpha_2 \) and \( \alpha_3 \) are weights for controlling the contribution of each pixel in the \( 3 \times 3 \) area. Using this \( 3 \times 3 \) kernel is supposed to give extra robustness against the subpixel accuracy and illumination changes [16,17,14,18]. In our experiments, all these weights are set as \( \alpha_1 = 1, \alpha_2 = 0.25 \) and \( \alpha_3 = 0.125 \) which refer to the distance from the centre pixel \( x \) of the area. This Error Score is intended to evaluate the optical flow at a specific pixel. We also use it to evaluate feature matching scores later in our framework.

**4. Step Two: Labeling Anchor Frames**

After obtaining our optical flow fields, anchor frames are then detected in a similar manner to Beeler et al. [2], with the difference that we employ SIFT for feature matching as opposed to Normalised Cross Correlation (NCC), and additionally use our Error Score function (Sec. 3) to evaluate matches. The main procedure is as follows (Fig. 3):

- **Feature Capture.** A set of SIFT features \( S_R \) is detected in the reference frame \( I_R \). Note that other features could be employed, but we select SIFT due to the general high accuracy and robustness. Here we apply the GPU version matching approach [9] to perform correspondence matching of SIFT feature sets \( S_R \) to feature set \( S_i \) of any other frames \( I_i \).

- **Outlier Rejection.** The aim of this selection process is removing outliers from our feature matching on all the frames. Correspondence matches
Fig. 3. Step Two. The frames are detected as anchor frames (Red) because of the similar appearance to the reference (Blue). These anchor frames partition the entire sequence into several independent clips which allows tracking performing in parallel.

Fig. 4. The anchor frames are selected based on our general error score which is computed by comparing the reference frame to every other frame in our Carton benchmark sequence.

of the SIFT feature set $S_R$ between the reference frame $I_R$ and the target frame $I_i$ are performed. We select the matches which meet $|\mathbf{x} - \mathbf{x'}| < \tau$ where $\mathbf{x}$ is feature position in $I_R$, $\{\mathbf{x} \in S_R, \mathbf{x} = (x, y)^T\}$; $\mathbf{x'}$ is the corresponding feature position in $I_i$; $\tau$ is a threshold which is set as 30 pixels in our experiments. We find this simple outlier rejection strategy sufficient for most of cases in our experiments (Sec. 7). More sophisticated outlier rejection method such as [22] could also be employed.

- **General Error Score.** The general error score is computed for every image as the average of the overall Error Score $E(\omega)$ (Eq. 1). Frames that contain the lowest general error score (below a specific threshold) are selected as anchor frames denoted $I_A$ and the other frames are non-anchor frames. It is because that the general error score is supposed to quantise the general appearance deformation where low score presents the small appearance change. Fig. 4 shows this process on our Carton benchmark sequence.

After labeling anchor frames that are visually similar to reference frame, these are used as a basis to partition the entire image sequence into several independent clips. This also allows computation in the next steps to be performed in parallel. In addition, the mesh $M_R$ is propagated from the reference frame $I_R$ to each anchor frame $I_A$ using SIFT matches and a direct optical flow field between them. More detail can be found in Sec. 6.1. The propagated mesh in an anchor frame is denoted $M_A = (V_A, E_A, F_A)$. Because of large-displacement motion between anchor frames, and the fact that many images in a deformable sequence may not return to a reference point, these alone are typically insufficient to provide reliable tracking. In the next section, the Anchor Patch concept will be introduced to overcome this issue.

5. Step Three: Labeling Anchor Patches

The motivation of the original Anchor Frame method [2] is to provide multiple Starting Points for tracking. Since error accumulates, the technique is intended to reduce overall error accumulation across long image sequences. However, as mentioned in the previous section, large displacement motion and complex motion may yield a fact that most images in a video sequence have significant visual differences from the reference frame.

The main observation in long image tracking is that local spatial patterns throughout a sequence may be repeated - i.e. part of a cloth might return to the same
position several times throughout a video. We take advantage of these repeating regions in order to track between shorter segments, and thus alleviate error accumulation. Apart from taking an entire image as anchor information, an Anchor Patch is defined as a set of individual vertices or a group of pixels in the non-reference frame (any other frame in the sequence), which are highly correspondent to a specific part of the reference. The benefit of using anchor patches is to provide additional information for correcting accumulated errors when tracking using optical flow. This technique can also reduce the impact of a low-quality anchor frame (i.e. the one is too dissimilar from the reference frame). Before anchoring patches on non-anchor frames, we first obtain a set of high-quality SIFT feature matches between the reference frame and anchor frames, we first obtain a set of high-quality SIFT feature matches between the reference frame and non-anchor frames, i.e. those frames are not already labelled as the reference frame, or an existing anchor frame. This process proceeds as follows:

- Feature Capture. In order to save the computational time, we reuse the SIFT feature sets from Step Two (Sec. 4). Here the SIFT feature set is denoted as $S_R$ in the reference frame $I_R$; $S_I$ presents a feature set of non-anchor frame $I_i$.

- Matching Selection. We also reuse the refined matchings from Step Two (Sec. 4). This process generates a matches set $m_{R \rightarrow i}$ from $S_R$ to $S_I$.

The set of matches $m_{R \rightarrow i}$ is used as our initial basis for anchoring patches on non-anchor frames. In order to obtain final anchor patches, Barycentric Coordinate Mapping and Error Refinement are applied as follows:

5.0.1. Barycentric Coordinate Mapping

We suppose to determine the pixel position in a non-anchor frame which corresponds to the position of a vertex on the reference mesh $M_R$ in $I_R$. These correspondences provide our baseline for stable tracking throughout the image sequence. Fig. 5 illustrates the process of anchoring patches where $v = (x, y)^T$ denotes a vertex in $M_R$; $f_\ast = (x_\ast, y_\ast)^T$, and denotes SIFT features in the reference frame $I_R$. Similarly, $f'_\ast = (x'_\ast, y'_\ast)^T$ denotes SIFT features in a non-anchor frame $I_i$. For the non-anchor frame $I_i$, we have $\{f_k \rightarrow f'_k \in m_{R \rightarrow i}, k = 1, 2, 3, \ldots\}$ which denotes previously obtained corresponding SIFT feature matches. We wish to calculate the new vertex position $v' = (x', y')^T$ in the non-anchor frame $I_i$. We do this by searching for the three nearest SIFT features $f_k$ in a small $5 \times 5$ search window centred on the vertex of interest $v$. Next, $v'$ is calculated by solving the Barycentric Coordinate Mapping equations as:

$$
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\beta_3
\end{bmatrix} = 
\begin{bmatrix}
f_1 \\
f_2 \\
f_3
\end{bmatrix} \cdot
\begin{bmatrix}
f'_1 \\
f'_2 \\
f'_3
\end{bmatrix}
$$

(2)

Where $\beta_{\ast}$ are intermediate variables that satisfy $\beta_1 + \beta_2 + \beta_3 = 1$. In practice we found this technique to provide an accurate transformation when applied to small region ($5 \times 5$ pixel block). However, more sophisticated (although slower) interpolation methods could also be used. The process is performed on every vertex in $M_R$.

5.0.2. Error Refinement

After Barycentric Coordinate Mapping, candidate anchor patches denoted by $v'_{\ast}$ are obtained in non-anchor frames $I_i$. We also have matches $v_{\ast} \rightarrow v'_{\ast}$, the strength of which can be evaluated using our error equation (1). Using this error, we select final anchor patches in a non-anchor frame $I_i$ using $\{P(v'_{\ast})|E(v_{\ast} \rightarrow v'_{\ast}) < \eta\}$ where $\eta$ is a predefined threshold.

6. Step Four: Mesh Propagation

The objective of our optimisation framework is to track a mesh $M_R$ from the reference frame to every other frame in an image sequence. Given tracking information from the previous sections, this process is separated into two steps: first, the mesh $M_R$ is propagated from reference frame to anchor frames (Sec. 4)
and [6.1]. Second, the propagated mesh $M_A$ is propagated from anchor frames to the non-anchor frames within the clip (Sec. [6.2]).

6.1. Propagating from the reference frame to anchor frames

The mesh propagation process from the reference frame to the anchor frame is as follows:

- **Computing the optical flow field.** The optical flow field $w_{R \rightarrow A}$ directly between the reference frame to the anchor frame is computed by sum up pairwise optical flow fields $w_{i \rightarrow i+1}$ in between.
- **Matching selection.** We propagate the whole mesh $M_R$ from the reference to the anchor frames. For every vertex in $M_R$, high error matches in anchor frames are eliminated (see Error Refinement).
- **Barycentric Coordinate Mapping.** The positions of those eliminated vertices are recomputed by applying Barycentric Coordinate Mapping to low error matches. The operation is shown in Fig. 6.

After this stage, information for every vertex in $M_R$ is established from the reference frame to the anchor frame.

6.2. Propagating from anchor frames to non-anchor frames

The entire image sequence is partitioned into clips which are bound by different anchor frames. The propagation process can be individually performed within these clips in parallel. Within these clips, the anchor patches are supposed to improve overall tracking stability and accuracy. In order to use anchor patches in this process, we define Nearest Anchor Patch as follows. For vertex $v$ in $M_A$, the Nearest Anchor Patch of $v$ on frame $I_i$ is the anchor patch $\{ v'_{i+k} | v \rightarrow v'_{i+k} \}$ on non-anchor frame $I_{i+k}$ which is nearest to $I_i$ in the image sequence. Fig. 8 shows an example where frame $I_{i+k}$ is the frame which is nearest to frame $I_i$ in image sequence and contains anchor patch $v'_{i+k}$ matching to $v$ in anchor frame $I_A$. The main tracking procedure proceeds (Fig. 7) as follows:

- **Mesh propagation.** In order to establish tracking information between anchor frames and non-anchor frame, the mesh $M_A$ is first propagated from anchor frame $I_A$ to non-anchor frames $I_i$ using the previously calculated optical flow field $w_{A \rightarrow i}$ from Step One (Sec. [3]).
- **Anchor patches propagation.** The Nearest Anchor Patch of each vertex $v$ in $M_A$ is searched through the whole clip then propagated to non-anchor frame $I_i$ using the optical flow field in the forward $w_{* \rightarrow i}$ or backward $w'_{i+k \rightarrow i}$ direction.
- **Conflict eliminating.** After propagating the mesh and nearest anchor patches to non-anchor frame $I_i$, there may be position conflict on some of the propagated vertices. As shown in Fig. 8, $\tilde{v}_i$ and $\tilde{v}'_i$ are not in the same desired position. In order to eliminate the conflict, the position of $\{ v_i | v \rightarrow v_i \}$ matching to $v$ can be calculated using the sum of all weighted candidate positions e.g. $\tilde{v}_i$ and $\tilde{v}'_i$ (Eq. 3) based on the Error Score.

$$ v_i = \frac{E(v \rightarrow \tilde{v}_i)\tilde{v}_i + E(v \rightarrow \tilde{v}'_i)\tilde{v}'_i}{E(v \rightarrow \tilde{v}_i) + E(v \rightarrow \tilde{v}'_i)} $$ (Eq. 3)

Due to the fact that the anchor frames divide the overall sequence into smaller clips, this allows the mesh propagation in between to be calculated in parallel. In the next section we perform an evaluation of our framework.

7. Evaluation

We evaluate APO with a range of 6 popular optical flow estimation methods which are publicly avail-
able from the Middlebury Evaluation System [1]. Combined local-global Optical Flow (CLG-TV) [3], Large Displacement Optical Flow (LDOF) [6] and Classic+NL [24] are state of the art while the Horn and Schunck (HS) [13], Black and Anandan (BA) [3,24], Improved TV-L1 (ITV-L1) [26] are classic optical flow frameworks and also widely used. CLG-TV is a high speed approach that uses a combination of bilateral filtering and anisotropic regularization and also one of the top three algorithms in the normalized interpolation error test from Middlebury. LDOF is an integration of rich feature descriptors and variational optical flow and one of best current optical flow estimation algorithms for large displacement motion. Classic+NL provides high performance in the Middlebury evaluation by formalizing the median filtering heuristic and Lorentzian penalty as explicit objective functions in an improved TV-L1 framework. The HS method is a pioneering technique optical flow. BA provides improvements to the HS framework by introducing robust quadratic error formulation. ITV-L1 is a recent and increasingly popular optical flow framework which uses a similar numerical optimisation scheme to Classic+NL. Our choice of a mixture of newer, state of the art methods, with older traditional approaches, is to highlight the fact that irrespective of the approach used, our APO framework provides significantly improved tracking in all cases.

For our evaluation, we compare the optical flow estimation methods previously mentioned – with and without our optimisation framework – on 7 long benchmark sequences with ground truth. Table 2 gives an overview of the benchmark sequences used in our evaluation. In previous work Garg et al. released to the community a set of ground truth data for evaluating optical flow algorithms over long sequences. This is as opposed to the Middlebury dataset, which just considers optical flow between pairs of images, and is therefore not applicable to our framework. The sequences of Garg et al. contains 60 frames and are generated using interpolated dense Motion Capture (MOCAP) data from real deformations of a waving flag [27]. As shown in Fig. 9 we use the same MOCAP data to generate a long video sequence and three other degraded sequences, each of which contains 237 frames of size 500 × 500 pixels. The three degraded sequences are generated in order to test the robustness of our APO framework under different image conditions. They are generated by individually adding synthetic occlusions, gaussian noise and salt & pepper noise with the same parameters described in [10]. In order to increase the diversity of the sequences, we include three other sequences. One is a Talking Face Video (Frank) sequence which contains 300 frames with 68 ground truth annotation points per frame. The other two are also synthetic benchmark sequences generated using MOCAP data of Salzmann et al. [23] from the carton and serviette deformations. One contains 266 frames of size 1024 × 768 while the other contains 307 frames of the same image size. In addition, we also consider the effect of the number of SIFT features detected in the frame, and how this affects overall tracking stability of the APO framework. All optical flow algorithms are applied with default parameter settings from their original papers.

Our baseline optical flow based tracking strategy – for each of the above algorithms – is performed as follows: First, the optical flow field is computed (in forward direction) for every pair of adjacent frames in the sequence. We then mark the initial tracking points in the first frame using the same ground truth points in the same frame of the sequence (Table 2). The corresponding points in the next frame are computed based on the optical flow field in between. This process is repeated until correspondent landmark points are obtained in every frame of the sequence. The aver-
Information of the Benchmark Sequences

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Original Occlusion</th>
<th>Guass Noise</th>
<th>S&amp;P Noise</th>
<th>Carton</th>
<th>Serviette</th>
<th>Frank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size (pix.)</td>
<td>500 × 500</td>
<td>500 × 500</td>
<td>500 × 500</td>
<td>500 × 500</td>
<td>1024 × 768</td>
<td>1024 × 768</td>
</tr>
<tr>
<td>Sequence Length</td>
<td>237</td>
<td>237</td>
<td>237</td>
<td>237</td>
<td>266</td>
<td>307</td>
</tr>
<tr>
<td>Annotation Points</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>81</td>
<td>63</td>
</tr>
<tr>
<td>Avg. Feature Amount</td>
<td>364.80</td>
<td>358.32</td>
<td>566.13</td>
<td>1276.50</td>
<td>2498.01</td>
<td>3315.49</td>
</tr>
</tbody>
</table>

Table 2

An overview of the benchmark sequences in our evaluation. That includes 4 attributes of image size (pixel), sequence length, number of ground truth annotation points per frame and average SIFT feature amount per frame.

Fig. 9. Our long nonrigid synthetic sequences with ground truth are rendered by using flag MOCAP data [27] and Garg et al. [11]. Images are adopted from [10].

age Endpoint Error (EE) [1] is then calculated against the ground truth annotation points. We then apply our APO framework using the same optical flow fields.). Note that the parameter values relevant to the APO framework are initially and experimentally selected, but then remain constant in all our evaluations.

Table 3(a) shows the measurement of average Endpoint Error (AEE) in pixels over all the frames of the sequences. We highlight the top three best AEE measures for each sequence using superscripts next to different values. Notice that APO significantly reduces the AEE compared to the baseline optical flow methods. Our optimisation framework yields the best AEE measure in all the cases. For instance, ITVL1 with APO performs the best in sequence Original while LDOF with APO yields the best result in sequence Frank. We also observe that although in the Guass.Noise and S&P.Noise sequences the improvement is less than in the unaltered sequences, the overall result is still an improvement with the addition of APO. We also observe that LDOF gives good results even without APO. It is because that the LDOF framework takes into account both regular optical flow energy and the feature technique. The latter contributes additional accuracy.

Table 3(b) shows another experiment, in which we performs Garg et al. [10] and Pizarro et al. [22] on our benchmark sequences (results on Carton, Serviette and Frank are not available) using a direct tracking strategy. Here we compute the optical flow fields directly from the reference to any other frames of the sequence. The annotation points are then directly tracked to the test frames using those flow fields. Note that the numbers in Table 3(b) may be slightly different from their original work [11]. It is because that, first, our sequences are extended to 237 frames which is around 3 times longer; second, we evaluate the tracking results of only 160 annotation points instead of all the pixels. We observe that the both state-of-the-art approaches (Garg et al. and Pizarro et al.) give higher accuracy than any other baseline methods in Table 3(a). The hidden conditions are (1) the tracking distance is minimum for Garg et al. and Pizarro et al. which very much reduces the accumulate errors; (2) both Garg et al. and Pizarro et al. shows high accuracy for nonrigid surface tracking in the record [10,22]. And all our sequences contain single nonrigid object. However, such direct tracking strategy cannot handle the situation where objects may be temporally out of the scene. In addition, the object appearance in the reference may be significantly different from the one in some other frames of the sequence. That brings extra difficulty to optical flow estimation. In this measure (Table 3(b)), our APO degrades to correct the tracking using the long term features. In the Original and Occlusion cases, the result shows that using APO still yields lower (or the same) errors over the baseline methods even those have given really minuscule measures (< 1.60 pixel). We also observe that APO slightly reduces the accuracy on the noisy cases because the extra noises affect the feature detection. Please note that the results are shown as "N/A" on Carton, Serviette and Frank as the baselines Garg at al. and Pizarro at al. are not available.
(a) Average Endpoint Error (AEE) comparison of different methods with our optimisation framework on the benchmark sequences.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Original</th>
<th>Occlusion</th>
<th>Gauss.N</th>
<th>S&amp;P.N</th>
<th>Carton</th>
<th>Serviette</th>
<th>Frank</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA [3]</td>
<td>6.14</td>
<td>8.03</td>
<td>11.02</td>
<td>7.79</td>
<td>10.56</td>
<td>5.18</td>
<td>17.57</td>
</tr>
<tr>
<td>BA + APO</td>
<td>1.72$^2$</td>
<td>1.91$^2$</td>
<td>$7.39^1$</td>
<td>$5.04^1$</td>
<td>2.77</td>
<td>$1.56^1$</td>
<td>6.60</td>
</tr>
<tr>
<td>CLG-TV [8]</td>
<td>8.59</td>
<td>10.93</td>
<td>20.28</td>
<td>33.93</td>
<td>28.94</td>
<td>32.17</td>
<td>19.29</td>
</tr>
<tr>
<td>CLG-TV + APO</td>
<td>2.25</td>
<td>2.97</td>
<td>12.31</td>
<td>18.99</td>
<td>6.95</td>
<td>9.43</td>
<td>7.05</td>
</tr>
<tr>
<td>HS [13]</td>
<td>29.16</td>
<td>30.44</td>
<td>29.74</td>
<td>29.43</td>
<td>27.69</td>
<td>37.90</td>
<td>31.27</td>
</tr>
<tr>
<td>HS + APO</td>
<td>11.68</td>
<td>12.88</td>
<td>17.79</td>
<td>17.21</td>
<td>10.25</td>
<td>10.03</td>
<td>14.19</td>
</tr>
<tr>
<td>LDOF + APO</td>
<td>1.75$^3$</td>
<td>$1.67^1$</td>
<td>11.65</td>
<td>13.12</td>
<td>$1.18^1$</td>
<td>$1.84^2$</td>
<td>$3.12^1$</td>
</tr>
<tr>
<td>Classic+NL + APO</td>
<td>2.15</td>
<td>3.18</td>
<td>8.31$^2$</td>
<td>6.46$^2$</td>
<td>1.34$^2$</td>
<td>2.03$^3$</td>
<td>3.44$^2$</td>
</tr>
<tr>
<td>ITV-L1 [26]</td>
<td>5.73</td>
<td>8.25</td>
<td>17.29</td>
<td>14.49</td>
<td>5.34</td>
<td>7.11</td>
<td>17.91</td>
</tr>
<tr>
<td>ITV-L1 + APO</td>
<td>$1.50^1$</td>
<td>2.33$^3$</td>
<td>9.53$^1$</td>
<td>7.70$^3$</td>
<td>$1.70^3$</td>
<td>2.36</td>
<td>3.69$^3$</td>
</tr>
</tbody>
</table>

(b) Average Endpoint Error (AEE) comparison of Garg et al. and Pizarro et al. on the benchmark sequences (directly tracking from the reference to any other frames).

<table>
<thead>
<tr>
<th>Methods</th>
<th>Original</th>
<th>Occlusion</th>
<th>Gauss.N</th>
<th>S&amp;P.N</th>
<th>Carton</th>
<th>Serviette</th>
<th>Frank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Garg et al., PCA [10]</td>
<td>0.61</td>
<td>0.71</td>
<td>1.64</td>
<td>1.21</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Garg et al., PCA [10] + APO</td>
<td>0.60</td>
<td>$0.69^1$</td>
<td>1.65</td>
<td>1.23</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Garg et al., DCT [10]</td>
<td>$0.59^1$</td>
<td>0.74</td>
<td>1.86</td>
<td>1.54</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Garg et al., DCT [10] + APO</td>
<td>$0.59^1$</td>
<td>0.73</td>
<td>1.85</td>
<td>1.53</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Pizarro et al. [22]</td>
<td>0.71</td>
<td>0.79</td>
<td>0.99</td>
<td>0.98</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Pizarro et al. [22] + APO</td>
<td>0.79</td>
<td>0.81</td>
<td>$0.99^1$</td>
<td>$0.98^1$</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 3
Average Endpoint Error (AEE) comparison on our long benchmark sequences.

While we consider ourselves primarily with tracking over long sequences, the shorter sequences are considered as well. In Table 4, the AEE measures of various methods are compared on the first 30 frames of our benchmark sequences. We observe similar AEE measures as in the long sequence case (Table 2(a)). The APO framework significantly increases the tracking accuracy – outperforming the baseline tracking methods in all cases even given degradation (e.g. Gauss.Noise and S&P.Noise). Moreover, the BA with APO is also observed to outperform in the noisy sequences while Classic+NL with APO yields the best measures in both sequences of Gauss.Noise and S&P.Noise.

We also evaluate the effect on tracking accuracy by varying the number of selected features. Different numbers (50% and 0%) of features are randomly selected from the initial full detection feature set before performing Anchor Patch detection. Information on our total number of features can be found in Table 2, e.g. there are 364.80 features averagely on each frame of the sequence Original. Table 5 shows an AEE comparison given various numbers of features. We observe that AEE is improved given more features in all cases. Another interesting observation is that our optimisation framework provides lower error against the baseline tracking strategy even given sparse or no features (0% feature). Note that in this case, our APO framework defaults to using an optical flow method with just the Anchor Frame approach [2]. Also note – for example by comparing to Table 3 – that this indicates that the APO framework also provides significant tracking improvement over using anchor frames alone.

We also make the visual comparisons on two of our sequences, Frank and Serviette. The former is a real world sequence with ground truth annotation points, while the latter is synthetic sequence overlaid with a ground truth mesh. In Fig. 10 we observe noticeable drift problems given the baseline optical flow tracking strategy. Also note that more details can be found in the corresponding video footage where we visually show that our framework significantly reduces the drift.
The computational consumption of our framework heavily relies on the supplementary optical flow method, because we need to calculate the optical flow fields twice (forward and backward) for every pair of adjacent images. Apart from this, our framework can be implemented in a parallel computation fashion. An-

<table>
<thead>
<tr>
<th>Methods</th>
<th>Average Endpoint Error (AEE) on the First 30 Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original Occlusion Guass.N S&amp;P.N Carton Serviette Frank</td>
</tr>
<tr>
<td>BA [1]</td>
<td>1.57 1.72 3.87 2.71 2.37 1.563 8.76</td>
</tr>
<tr>
<td>BA + APO</td>
<td>1.413 1.65 3.663 2.132 2.17 1.133 5.40</td>
</tr>
<tr>
<td>CLG-TV [3]</td>
<td>2.40 2.60 6.71 8.77 8.10 5.54 8.60</td>
</tr>
<tr>
<td>CLG-TV + APO</td>
<td>2.10 2.24 6.53 8.39 4.79 5.11 7.35</td>
</tr>
<tr>
<td>HS [13]</td>
<td>33.67 35.70 35.05 34.50 26.16 22.08 12.76</td>
</tr>
<tr>
<td>LDOF [6]</td>
<td>2.38 2.37 3.96 4.03 3.90 2.52 8.51</td>
</tr>
<tr>
<td>LDOF + APO</td>
<td>1.152 0.971 3.75 2.66 0.891 1.442 2.821</td>
</tr>
<tr>
<td>Classic+NL [24]</td>
<td>1.63 1.76 3.612 2.513 2.18 1.75 8.77</td>
</tr>
<tr>
<td>Classic+NL + APO</td>
<td>1.51 1.332 3.541 1.991 1.242 1.68 3.703</td>
</tr>
<tr>
<td>ITV-L1 [26]</td>
<td>1.55 1.76 6.27 5.07 2.37 2.01 9.22</td>
</tr>
<tr>
<td>ITV-L1 + APO</td>
<td>0.991 1.312 5.77 4.65 1.693 1.71 3.482</td>
</tr>
</tbody>
</table>

Table 4

Average Endpoint Error (AEE) comparison of different methods with our optimisation framework on the first 30 frames of the benchmark sequences.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Average Endpoint Error (AEE) on Different Feature Distributions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original Occlusion Guass.N S&amp;P.N Carton Serviette Frank</td>
</tr>
<tr>
<td>BA [1], No APO</td>
<td>6.14 8.03 11.02 7.79 10.56 5.18 17.57</td>
</tr>
<tr>
<td>APO, 100% Feature</td>
<td>1.722 1.912 7.891 5.041 2.77 1.561 6.60</td>
</tr>
<tr>
<td>APO, 50% Feature</td>
<td>3.64 4.71 8.06 6.12 5.89 2.98 10.63</td>
</tr>
<tr>
<td>APO, 0% Feature</td>
<td>5.12 6.44 9.23 7.21 8.69 4.35 12.69</td>
</tr>
<tr>
<td>CLG-TV [3], No APO</td>
<td>8.59 10.93 20.28 33.93 28.94 32.17 19.29</td>
</tr>
<tr>
<td>APO, 100% Feature</td>
<td>2.25 2.97 12.31 18.99 6.95 9.43 7.05</td>
</tr>
<tr>
<td>APO, 50% Feature</td>
<td>4.86 6.51 14.39 22.72 15.36 19.91 12.00</td>
</tr>
<tr>
<td>APO, 0% Feature</td>
<td>6.94 9.11 16.83 26.03 23.57 24.03 15.07</td>
</tr>
<tr>
<td>HS [13], No APO</td>
<td>29.16 30.44 29.74 29.43 27.69 37.90 31.27</td>
</tr>
<tr>
<td>APO, 100% Feature</td>
<td>11.68 12.88 17.79 17.21 10.25 10.03 14.19</td>
</tr>
<tr>
<td>APO, 50% Feature</td>
<td>18.13 20.28 20.66 19.91 17.39 25.99 23.45</td>
</tr>
<tr>
<td>APO, 0% Feature</td>
<td>24.73 27.11 23.97 23.40 24.09 33.11 29.17</td>
</tr>
<tr>
<td>LDOF [6], No APO</td>
<td>6.21 6.39 16.24 24.14 6.33 5.51 14.73</td>
</tr>
<tr>
<td>APO, 100% Feature</td>
<td>1.753 1.671 11.65 13.12 1.181 1.842 3.121</td>
</tr>
<tr>
<td>APO, 50% Feature</td>
<td>3.21 3.09 12.18 15.02 2.90 3.74 8.66</td>
</tr>
<tr>
<td>APO, 0% Feature</td>
<td>5.08 5.24 14.11 18.46 5.45 4.89 11.76</td>
</tr>
<tr>
<td>Classic+NL [24], No APO</td>
<td>7.07 10.61 12.65 9.50 5.72 6.62 17.32</td>
</tr>
<tr>
<td>APO, 100% Feature</td>
<td>2.15 3.18 8.312 6.462 1.342 2.033 3.442</td>
</tr>
<tr>
<td>APO, 50% Feature</td>
<td>4.00 6.39 9.48 7.33 3.89 4.00 10.14</td>
</tr>
<tr>
<td>APO, 0% Feature</td>
<td>5.96 7.78 11.64 8.98 4.78 6.00 13.27</td>
</tr>
<tr>
<td>ITV-L1 [26], No APO</td>
<td>5.73 8.25 17.29 14.49 5.34 7.11 17.91</td>
</tr>
<tr>
<td>APO, 100% Feature</td>
<td>1.503 2.333 9.533 7.703 1.703 2.36 3.693</td>
</tr>
<tr>
<td>APO, 50% Feature</td>
<td>3.59 5.17 10.93 8.47 3.41 5.00 10.11</td>
</tr>
<tr>
<td>APO, 0% Feature</td>
<td>4.77 6.92 12.50 10.31 4.43 5.95 14.29</td>
</tr>
</tbody>
</table>

Table 5

Average Endpoint Error (AEE) comparison on the benchmark sequences with varying feature distributions.
chor frames divide the sequence into clips which give multiple start points for tracking. In the implementation, a GPU version of SIFT approach [9] is applied for feature detection and matching (around 10 frames per second on our benchmarks). The whole framework is constructed under CUDA platform. Assuming all optical flow fields are obtained, our framework reach real-time efficiency (around 2 frames per second) on our benchmarks using on a 2.9Ghz Xeon 8-cores, NVIDIA Quadro FX 580, 16Gb memory computer.

8. Conclusion

In this paper, we have presented a novel optimisation framework using Anchor Patches constraint, which improves the tracking on mesh or sparse points through long image sequences. Our optimisation framework temporally anchors the image regions throughout the sequence in order to mitigate the effect of Error Accumulation (Drift). In the evaluation, our approach combined with 6 popular optical flow algorithms and show significant improvement against baselines methods on 7 benchmark sequences. Such datasets include 6 synthetic benchmark sequences with realworld deformation and 1 realworld sequence.

9. Acknowledgements

We thank Ravi Garg and Lourdes Agapito for providing their GT datasets. We also thank Gabriel Brostow and the UCL Vision Group for their generous comments. The authors are supported by the EPSRC CDE EP/L016540/1 and CAMERA EP/M023281/1; and EPSRC projects EP/K023578/1 and EP/K02339X/1.

References


(a) Visual comparison of different methods on the frame 88 of the sequence *Frank*.

(b) Visual comparison of different methods on the frame 192 of the sequence *Serviette*.


2015.