Combining local and global motion models for feature point tracking

Aeron Buchanan
Dept. Engineering Science
University of Oxford, UK
amb@robots.ox.ac.uk

Andrew Fitzgibbon
Microsoft Research
Cambridge, UK
http://research.microsoft.com/~awf

Abstract

Accurate feature point tracks through long sequences are a valuable substrate for many computer vision applications, e.g. *non-rigid body tracking, video segmentation, video matching, and even object recognition.*

Existing algorithms may be arranged along an axis indicating how global the motion model used to constrain tracks is. Local methods, such as the KLT tracker, depend on local models of feature appearance, and are easily distracted by occlusions, repeated structure, and image noise. This leads to short tracks, many of which are incorrect. Alone, these require considerable postprocessing to obtain a useful result. In restricted scenes, for example a rigid scene through which a camera is moving, such postprocessing can make use of global motion models to allow "guided matching" which yields long high-quality feature tracks. However, many scenes of interest contain multiple motions or significant non-rigid deformations which mean that guided matching cannot be applied.

In this paper we propose a general amalgam of local and global models to improve tracking even in these difficult cases. By viewing rank-constrained tracking as a probabilistic model of 2D tracks rather than 3D motion, we obtain a strong, robust motion prior, derived from the global motion in the scene. The result is a simple and powerful prior whose strength is easily tuned, enabling its use in any existing tracking algorithm.

1. Introduction

Feature point tracking plays many foundational roles in video processing, *e.g*. cinema post-production, non-rigid body tracking [8], video segmentation [4], video matching [11], and even object recognition [13]. As such, accurate and efficient approaches that provide precise tracking data over long sequences are desirable, both to minimize user input in manual tasks and make automated systems more effective. Most trackers are built on top of local feature-point trackers such as Harris or difference-of-Gaussian interest point detection and template or SIFT descriptor search [2, 9] or the Kanade-Lucas-Tomasi (KLT) tracker [10]. However, feature tracking through image sequences of general motion remains a difficult problem—repeating textures, occlusions and appearance change can frustrate even the most sophisticated tracking algorithms.

The problem is greatly simplified when tracking in scenes for which a global motion model is available, *e.g*. when it is a rigid, unchanging world that is being filmed. In such cases, "guided matching" using the motion model allows accurate tracks to be obtained, even in the presence of the above difficulties [2, 6]. Beardsley *et al*. [2] showed how the assumption that the camera is passing through a rigid 3D world greatly improves 2D point matching. Bregler *et al*. [3] showed how the rigidity assumption could be generalized to include non-rigid deformation, while Irani [6] showed how hard global motion constraints can be incorporated into direct optical flow methods for a range of rigid motion models. Torresani *et al*. [15] extend and combine these techniques and describe an automatic process for simultaneously tracking sets of feature points and fitting a non-rigid model to constrain the motion.

All of the above methods share the same disadvantage: because the global motion model must apply to the whole sequence, long sequences containing complicated non-rigid or multiple body motion must be described by a very general motion model which cannot reliably constrain the feature tracks. If a more specific motion model is used, tracks on small moving objects will not be obtainable.

There is also a class of semi-local models. Sand and Teller [12] simultaneously compute the trajectory of a large set of image features under a piecewise smoothness model implemented by first computing flow vectors under a gradient-weighted smoothness constraint, and then bilateral filtering of the flow. Similarly, Smith *et al*. [14] refine matches using a median flow filter, again imposing a form of piecewise smoothness. These methods, although smooth globally, can exacerbate the feature drift problems of purely local methods; and lack the predictive power of the global

1-4244-1180-7/07/$25.00 ©2007 IEEE

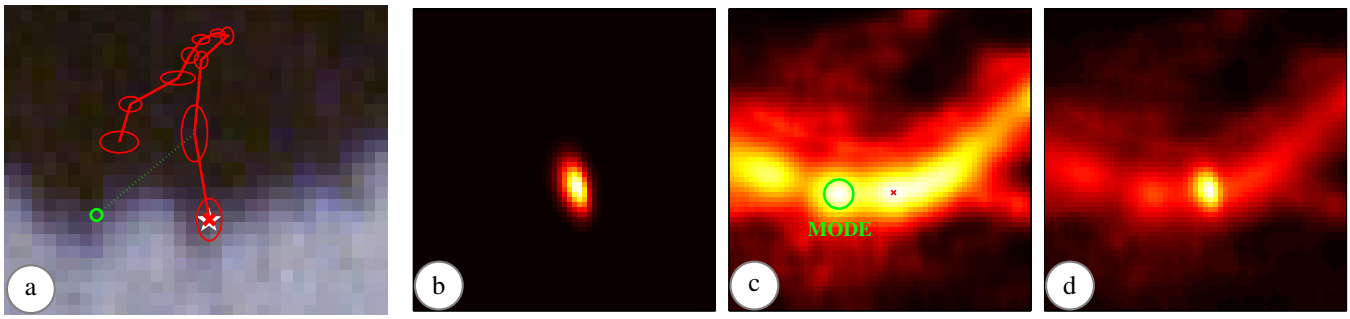

Figure 1. **Bayesian tracking with motion prior**. (a) Frame 38 from the giraffe sequence (see Figure 3) with the feature track (plus covariance ellipses from Equation 14) over the previous 9 frames (used to calculate the prior) superimposed. The white star is the ground truth. The green circle is the naïve maximum likelihood estimate, confused by the repeated texture. (b) The diffused prior $\mathrm{q}(\mathbf{x}_t) = \mathrm{p}(\mathbf{x}_t|\mathbf{z}_{t-\mathtt{W}})$. (c) The likelihood function over position $\mathrm{p}(\mathbf{z}_t|\mathbf{x}_t)$. Note that the mode is far from the correct peak. (d) The posterior $\mathrm{p}(\mathbf{x}_t|\mathbf{z}_{t,t-\mathtt{W}})$ clearly showing that the motion model has meant the erroneous location is avoided and the correct point will be chosen.

methods, where trajectories in previous frames can be used to predict feature positions in future frames.

This paper combines local and global models by continuously updating a non-rigid motion model over a sliding temporal window (typically of the order of 10 frames). Then when tracking an individual feature, this motion model is used as a motion prior in a conventional Bayesian template tracker. As such we draw on the strong information held in the local optical flow information and the weaker global motion information used by all the methods described above. We show that the model can track long sequences with significant appearance variation, lighting changes and motion blur.

2. Description

The task is to return an accurate feature point trajectory through an image sequence of a scene undergoing general motion, including multiple bodies and non-rigidity, given the location of the feature in a single starting frame. For each frame, t, we wish to know the underlying position $\mathbf{x}_t$ of the feature, given the observations of the images, $\mathtt{I}_{1:t}$, in the current and all previous frames. The standard Bayesian tracking formulation [1] poses the search for the probability density over position as

$$\underbrace{\mathrm{p}(\mathbf{x}_t|\mathtt{I}_{1:t})}_{\text{posterior}} \propto \underbrace{\mathrm{p}(\mathtt{I}_t|\mathbf{x}_t)}_{\text{likelihood}} \int \underbrace{\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-1})}_{\text{motion model}} \underbrace{\mathrm{p}(\mathbf{x}_{t-1}|\mathtt{I}_{1:t-1})}_{\text{prior}} \mathrm{d}\mathbf{x}_{t-1} \tag{1}$$

with the posterior at time t becoming the prior at time $t+1$. In addition, $\mathbf{x}$ is associated with an appearance model, for example a template patch, with parameters $\boldsymbol{\theta}$, so the above should be written replacing all instances of $\mathbf{x}$ with $(\mathbf{x}, \boldsymbol{\theta})$. There is considerable research in how to maintain and update the appearance density [7, 10], but in this paper we are primarily concerned with the density over position. Thus we shall adopt a very simple model of appearance, i.e. a normalized sum-of-squared differences (NSSD) match to the previous frame. Eliding the formal derivation, this modifies the likelihood as follows:

$$\mathrm{p}(\mathtt{I}_t|\mathbf{x}_t) \;\propto\; \exp\left(-\sigma^{-2}\,\mathrm{nssd}(\mathit{Wnd}(\mathtt{I}_t, \mathbf{x}_t), \boldsymbol{\theta}_{t-1})\right) \tag{2}$$

where the function $\mathit{Wnd}(\mathtt{I}, \mathbf{x})$ extracts a square window from a color image $\mathtt{I}$ centered at location $\mathbf{x}$. The template $\boldsymbol{\theta}_{t-1}$ is $\mathit{Wnd}(\mathtt{I}_{t-1}, \mathbf{z}_{t-1})$ where $\mathbf{z}_{t-1}$ is the KLT update of the mode of the prior, i.e. given the mode of the prior $\hat{\mathbf{x}}_{t-1}$ defined as

$$\hat{\mathbf{x}}_{t-1} = \operatorname*{argmax}_{\mathbf{x}_{t-1}} \mathrm{p}(\mathbf{x}_{t-1}|\mathtt{I}_{1:t-1}), \tag{3}$$

then a window around $\hat{\mathbf{x}}_{t-1}$ is extracted and used as the basis for a single-frame KLT update [10]. The 2D position to which this converges becomes $\mathbf{z}_{t-1}$, and throughout the rest of the paper the 2×1 vector $\mathbf{z}$ will be treated as an image feature. Finally, we extend the standard formulation and treat the motion as an order M Markov process, *i.e.* the motion model depends on a temporal window of $M-1$ frames. Therefore, we rewrite (1) in terms of $\mathbf{z}$ and a vector of time indices $\mathtt{W} = [1 \ldots M-1]$ as

$$\mathrm{p}(\mathbf{x}_t|\mathbf{z}_{t,t-\mathtt{W}}) \propto \mathrm{p}(\mathbf{z}_t|\mathbf{x}_t) \int \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-\mathtt{W}}) \mathrm{p}(\mathbf{x}_{t-\mathtt{W}}|\mathbf{z}_{t-\mathtt{W}}) \mathrm{d}\mathbf{x}_{t-\mathtt{W}}, \tag{4}$$

where the notation $\mathbf{z}_{t-\mathtt{W}}$ denotes the column vector $[\mathbf{z}_{t-1}^\top \ldots \mathbf{z}_{t-M+1}^\top]^\top$. This deviates from usual Bayesian tracking in that the prior must be re-estimated every time step. However, as we shall see in §2.3, this is a very straightforward calculation. For emphasis, we note again that the variables $\mathbf{z}$ are concrete position observations, obtained by KLT updates, of the hidden random variables $\mathbf{x}$, probability densities over which are maintained through tracking.

[1] First appearances of symbols are highlighted in marginal notes.

2.1. Algorithm overview

Before developing the theory further, it is perhaps useful to give an overview of the algorithm that will emerge. The algorithm follows the "guided matching" framework, where motion models are first computed for sub-sequences and then used to constrain tracking on a second pass. The main steps are as follows.

First pass: Fit motion models.

1. Detect interest points on each frame and match to interest points in successive frames to generate tracks through the sequence.
2. Robustly fit a rank R non-rigid motion model to the tracks in each (overlapping) M-frame subsequence. In all our examples, we have used $M = 10$ and $R = 6$.

Second pass: For a single track, initialize the track by standard KLT tracking for $M-1$ frames forming observations $\mathbf{z}_{1..R}$. Now for each subsequent frame t,

1. Predict the position in frame t as the *diffused prior distribution* $\mathrm{q}(\mathbf{x}_t) = \int \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-\mathtt{W}})\mathrm{p}(\mathbf{x}_{t-\mathtt{W}}|\mathbf{z}_{t-\mathtt{W}})\mathrm{d}\mathbf{x}_{t-\mathtt{W}}$.
2. Measure the likelihood $\mathrm{p}(\mathbf{z}_t|\mathbf{x}_t)$. Conceptually this would mean starting a KLT update from every point in the image, but in practice is implemented more efficiently by considering only local maxima of the NSSD cost surface.
3. Choose $\mathbf{z}_t$ as the KLT update of the point which maximizes the product of likelihood and diffused prior.
4. Approximate the posterior with a Gaussian.

Thus the basic KLT tracker is constrained by the global motion model $\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-\mathtt{W}})$, reducing the likelihood of drift and of jumping to incorrect matches. Because the global model is based on a sliding temporal window it can model complex motions. We now describe the key algorithm steps in more detail.

2.2. First pass: fitting the motion model

The motion model we employ is the non-rigid factorization model of Bregler [3] and Irani [6]. Briefly, given a sequence of M images, a tracked point is represented by a $2M$ long column vector $\mathbf{X} = [x_M, y_M, ..., x_1, y_1]^\top$. Given N tracked points in the sequence, the column vectors are concatenated horizontally into a $2M \times N$ *measurement matrix* $\mathtt{M}$, holding each track, $\mathbf{X}_i$, in a separate column and the positions of all the features in a particular frame in consecutive pairs of rows.

Our motion model takes the form of a basis that describes the motion seen in $\mathtt{M}$. This basis, $\mathtt{B}$, is a $2M \times R$ matrix *i.e.* it assumes that all scene motion for the M-frame sequence exists in a rank R subspace, that is $\mathtt{M} = \mathtt{BC}$ for $\mathtt{C}$ an $R \times N$ matrix of coefficients. It is important to note that $\mathtt{B}$ is obtained from the direct factorization of $\mathtt{M}$, *i.e.* without mean-centering, thus providing a complex and desirable spatial consistency for predictions. The rank, R, is the main tuning parameter of our algorithm chosen to represent the complexity of motion in the scene (avoiding overfitting).

The prior used in the second pass is based on the robust generation of this global motion model using standard template tracking of interest points over short subsequences of the long input sequence. We build measurement matrices $\mathtt{M}$ for all M-frame subsequences of the sequence. It is acknowledged that $\mathtt{M}$ will contain incomplete and erroneous tracks. However, this will be dealt with robustly, as described in §3.1. The output of this stage is a $2M \times R$ basis matrix $\mathtt{B}_t$ for each frame of the sequence, with $\mathtt{B}_t$ modelling the motion for frames from t down to $t-M+1$.

2.3. Second pass: Computing $\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-\mathtt{W}})$

In order to compute the prediction of position for the current frame, we use the motion basis $\mathtt{B}_t$, which for the rest of this section and the next we shall denote by $\mathtt{B}$. Concatenating the prediction $\mathbf{x}_t$ and history $\mathbf{x}_{t-\mathtt{W}}$ into a single $2M \times 1$ column vector, we obtain

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-\mathtt{W}} \end{bmatrix} = \mathtt{B}\mathbf{c} = \begin{bmatrix} \mathtt{B}_0 \\ \mathtt{B}_\mathtt{W} \end{bmatrix} \mathbf{c} \tag{5}$$

where $\mathtt{B}_0$ is the first two rows of $\mathtt{B}$, and $\mathtt{B}_\mathtt{W}$ denotes the remaining rows. The coefficient vector $\mathbf{c}$ may be computed from $\mathbf{x}_{t-\mathtt{W}}$ as

$$\mathbf{c} = \mathtt{B}_\mathtt{W}^+ \mathbf{x}_{t-\mathtt{W}} \tag{6}$$

where $\mathtt{B}_\mathtt{W}^+$ is the pseudoinverse of $\mathtt{B}_\mathtt{W}$, and thus the prediction of $\mathbf{x}_t$ is

$$\langle \mathbf{x}_t \rangle = \mathtt{B}_0 \mathtt{B}_\mathtt{W}^+ \mathbf{x}_{t-\mathtt{W}} = \mathtt{P}\mathbf{x}_{t-\mathtt{W}} \tag{7}$$

where the $2 \times 2(M-1)$ matrix $\mathtt{P} = \mathtt{B}_0 \mathtt{B}_\mathtt{W}^+$ projects the track $\mathbf{x}_{t-\mathtt{W}}$ into frame t. Even though this is a simple linear projection, it can model complex non-rigid motions (including sharp cusps) because of the use of the long time history and global support. We then define the distribution as

$$\mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-\mathtt{W}}) = \exp\left(-\gamma \|\mathbf{x}_t - \mathtt{P}\mathbf{x}_{t-\mathtt{W}}\|^2\right) \tag{8}$$

$$\equiv \mathcal{N}(\mathbf{x}_t \mid \mathtt{P}\mathbf{x}_{t-\mathtt{W}},\ \gamma^{-1}\mathtt{I}) \tag{9}$$

where γ is a tuning parameter called the *diffusion coefficient*. Large values of γ mean high confidence in the prediction, low values mean that tracking tends not to trust the motion model. We set $\gamma = 10$ for all tests.

2.4. Computing the diffused prior

We now wish to compute the diffused prior $\mathrm{q}(\mathbf{x}_t)$ over position, which will give the search window in image t.

Section 2.3 defines the predicted position of $\mathbf{x}_t$ given the previous positions $\mathbf{x}_{t-\mathtt{W}}$. We recall, however, that the $\mathbf{x}_{t-\mathtt{W}}$ are hidden variables, and that what was observed were the 2D positions $\mathbf{z}_{t-\mathtt{W}}$.

As will be discussed in §2.5, the prior will be expressed as a single Gaussian at the end of each tracking iteration. The mean of the Gaussian is $\mathbf{z}_{t-\mathtt{W}}$, and let $\Sigma_{t-\mathtt{W}}$ be the $2(M-1)\times 2(M-1)$ covariance matrix (discussed below). Thus

$$\mathrm{p}(\mathbf{x}_{t-\mathtt{W}}|\mathbf{z}_{t-\mathtt{W}}) = \mathcal{N}(\mathbf{x}_{t-\mathtt{W}} \mid \mathbf{z}_{t-\mathtt{W}},\ \Sigma_{t-\mathtt{W}}) \tag{10}$$

and the integral which gives the diffused prior over position in the new frame is

$$\mathrm{q}(\mathbf{x}_t) = \int \mathrm{p}(\mathbf{x}_t|\mathbf{x}_{t-\mathtt{W}})\mathrm{p}(\mathbf{x}_{t-\mathtt{W}}|\mathbf{z}_{t-\mathtt{W}})\mathrm{d}\mathbf{x}_{t-\mathtt{W}}. \tag{11}$$

Substituting (10) and (9), we obtain

$$\mathrm{q}(\mathbf{x}_t) \propto \int \exp(-\gamma\|\mathbf{x}_t - \mathtt{P}\mathbf{u}\|^2)\times \exp(-(\mathbf{u}-\mathbf{z}_{t-\mathtt{W}})^\top\Sigma_{t-\mathtt{W}}^{-1}(\mathbf{u}-\mathbf{z}_{t-\mathtt{W}})\mathrm{d}\mathbf{u} \tag{12}$$

where the variable of integration has been written $\mathbf{u}$ to aid legibility. Evaluating the integral (see appendix) yields a normal distribution for $\mathbf{x}_t$ of the form

$$\mathrm{q}(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t \mid \mathtt{P}\mathbf{z}_{t-\mathtt{W}},\ \gamma^{-1}\mathtt{I} + \mathtt{P}\Sigma_{t-\mathtt{W}}\mathtt{P}^\top). \tag{13}$$

2.5. Posterior update

Multiplying the likelihood computed from (2) by the diffused prior as above gives a posterior response surface $S(\mathbf{x}) = p(\mathtt{I}_t|\mathbf{x})\mathrm{q}(\mathbf{x})$. The posterior distribution $p(\mathbf{x}_t|\mathbf{z}_t)$ is then approximated by a Gaussian with mean given by the KLT update of the mode of the posterior as in Equation (3) and a diagonal covariance matrix $\Sigma_t = d^2\mathtt{I}_{2\times 2}$ with d heuristically set to the L_1 distance between the mode of $\mathrm{q}(\mathbf{x}_t)$ and $\mathbf{z}_t$. Thus the covariance of the full posterior is simply the $2M\times 2M$ block diagonal matrix

$$\Sigma_{t,t-\mathtt{W}} = \begin{bmatrix} \Sigma_t & \mathtt{0} \\ \mathtt{0} & \Sigma_{t-\mathtt{W}} \end{bmatrix}. \tag{14}$$

3. Implementation details

The above steps describe the essential components of our tracking algorithm. The main novelty is in the use of rank constraints to support a long-range (10-frame) motion model which allows accurate track predictions even with complex motion. In this section we go through some of the implementation strategies employed in building a robust system.

3.1. RANSAC for subsequence motions

In the first pass, the main task is to compute the best rank-R basis for a measurement matrix $\mathtt{M}$. For an $\mathtt{M}$ containing no incorrect tracks, the optimal rank R basis is obtained by factorization (*e.g.* singular value decomposition, SVD) and truncation to rank R. If occlusions and false positives have been recorded then a robust factorization algorithm should be used. In the rest of this section we describe a fast RANSAC-based algorithm which we have found to work well in practice.

Any robust batch tracking algorithm may be employed to generate the $\mathtt{M}$ for each subsequence, but faster and simpler approaches are more more desirable for maximizing throughput and the ability to capture arbitrary motion respectively. We have chosen to use a straightforward Harris-based batch tracker to create the $\mathtt{M}$s, followed by a RANSAC scheme to obtain faithful bases. The steps of the RANSAC stage are: pick R complete columns of the measurement matrix to form a candidate basis, $\mathtt{B}'$; calculate the reprojection error for all tracks in $\mathtt{M}$ using $\mathrm{e}(\mathbf{x}) = \|(\mathtt{I} - \mathtt{B}'\mathtt{B}'^{+})\mathbf{x}\|$; count the support as the number of tracks (considering only complete columns at this stage) with a reprojection error less than a threshold; choose the candidate basis with the largest support; calculate the best basis, $\mathtt{B}$, from all the tracks in the support for the best candidate $\mathtt{B}'$ using SVD and rank truncation.

The above RANSAC process alone tends to be too aggressive and leaves out many correct tracks. To further improve the quality of the basis for each subsequence, we employ a "basis growing" procedure, expanding it to include as many of the inliers as possible in a simple, yet surprisingly effective way. The algorithm is as follows. Until the number of tracks in the support stops increasing, recalculate the reprojection error of all the tracks using the new basis and reclassify them all (as inliers or outliers), using a threshold, to generate a new support. Calculate the new basis by rank truncation as above and iterate. A variation that can be slightly more conservative is to reduce the threshold to that of the worst reprojection error of the support in any given iteration.

If the sequence needs to be re-tracked using a different rank constraint, the same measurement matrices are used to generate new bases. The time taken to robustly find and grow a basis for a subsequence takes about the same amount of time as loading an image from disk.

3.2. Multiple predictions

To get a more robust motion estimate, we actually employ a range of bases to get a series of predictions for each point in each frame, using the range $M = \{6, 7, 8, 9, 10\}$ for $R = 6$. Values of $2M$ too close to the rank generally give motion predictions that are too erratic; making over-

constrained estimations tends to be more effective. However, considering previous motion over too many frames leads to the low rank motion approximation breaking down, hence the balance. The posterior is then generated from the mixture of the resulting Gaussians.

3.3. KLT refinement

As the track proceeds, we maintain a set of warp parameters, $\mathcal{P}$, holding the affine transformation of the starting feature's appearance, matching it to the current frame. It is initially set to the identity matrix, *i.e.* no warp. Starting at the current estimate location, given by maximizing the posterior estimate, and the current warp, $\mathcal{P}$, the local minimum NSSD fit of the original template to the current image is found. $\mathcal{P}$ is updated for the next frame by blending it with the new warp found for this frame, $\mathcal{P}'$. Because we expect little or no scaling, the blend is based on the eigenvalues of both the current and the new affine warp matrices. Furthermore, as parameter drift is a real threat in the cumulative update paradigm, we counteract the potential of the KLT fitting process taking the warp to extreme distortions by using the absolute difference in appearance between the original template and the warped match in the current frame. This pixel intensity error modulates a blend of the updated $\mathcal{P}$ with the identity matrix and so provides a helpful restraining influence on run-away optimizations.

3.4. Robustifying $\mathrm{q}(\mathbf{x}_t)$

In the form in (9), too much confidence is placed in the prediction of the motion model for practical use, so we implement a robust prior. We scale the prior to have maximum value one and make a mixture with a uniform distribution. The more robust prior, denoted q', is then defined by

$$\mathrm{q}'(\mathbf{x}_t) \propto \alpha \frac{\mathrm{q}(\mathbf{x}_t)}{\max_{\mathbf{x}} \mathrm{q}(\mathbf{x})} + (1-\alpha). \tag{15}$$

The blend coefficient, α, reflects the confidence we have in the model's predictive powers for the trajectory $\mathbf{z}_{t-\mathtt{W}}$. It can be quantified by comparing the coefficient vector $\mathbf{c}$ of the preceding trajectory $\mathbf{z}_{t-\mathtt{W}}$, given by $\mathbf{c} = \mathtt{B}^{+}\mathbf{z}_{t-\mathtt{W}}$, to the coefficients of all the inlying motion in the measurement matrix, $\mathtt{M}$. If the motion of the tracked feature currently matches that seen in the rest of the scene, then the prediction made by the basis can be taken as being good, hence

$$\alpha = e^{-\beta d_{min}} \tag{16}$$

for a scaling parameter β controlling the speed at which α decays with distance (we set $\beta = 0.0005$) and

$$d_{min} = \min_i(\|\mathbf{c} - \mathbf{c}_i\|_2) \tag{17}$$

where $\mathbf{c}_i$ is the i^{th} column of $\mathtt{B}^{+}\mathtt{X}$, for $\mathtt{X}$ the 'measurement matrix' of inliers used to determine $\mathtt{B}$. Effectively, this is a nearest neighbor estimate of the density of the basis coefficients.

When using multiple predictions, these α values are useful as weights for when the predictions are combined. We use them when averaging the modes of $\mathrm{q}(\mathbf{x}_t^i)$ for the calculation of d in Equation (14).

Name	Resolution	Length	Texture	Objects	Tracks
Giraffe	720×576	100	med	1 (N)	9
Leopard	700×475	242	high	2 (N+N)	9
Mouth	720×480	346	low	2 (N+R)	6
Zebras	720×576	171	high	10 (9N+R)	9

Table 1. **Ground-truth test sequences.** "Resolution" is in pixels. "Length" is measured in frames. "Texture" indicates the density of texture in the scene. "Objects" is the number of independently moving objects in the sequence as a whole (N: non-rigid, R: rigid). "Tracks" is the number of ground-truth tracks evaluated on each.

4. Experiments

We obtained ground truth for tracking on four sample image sequences, summarized in Table 1, and illustrated in Figure 3. Tracking challenges in the sequences include appearance variation, lighting changes and considerable motion blur.

As the paper's main contribution is in the form of the prior, we ran, along with the tracker described, the same Bayesian KLT tracker employing three other functions for calculating the prior:

Uniform. A uniform prior over a constant-size search window (61×61 pixels).

Acceleration. A constant acceleration model (covariances as in §2.5) whose motion parameters are re-estimated every frame using the last three observations.

Median. The median two-frame global motion within a 30 pixel radius [14].

The first experiment measures tracker reliability. Throughout the testing, 13×13 image patch templates were used and rank-6 motion bases were employed. We started the trackers on the ground truth track positions in image 1 of each sequence. When the track drifted off position by more than Δ pixels, a track failure was recorded. This was then repeated, starting the trackers in each of the first 100 frames of the sequences (first 50 for the shorter giraffe sequence) in order to average out any artifacts that may occur due to starting in any particular frame. This average track length is an important predictor of performance on many tracking tasks (*e.g.* structure and motion recovery [5]). Table 2 summarizes the average track length improvements of our proposed motion prior compared to the three models described above for $\Delta = 4$. Average track lengths increased in

Proposed vs.	Giraffe	Leopard	Mouth	Zebras
Uniform	**46.9%**	**80.9%**	**44.8%**	**16.0%**
Acceleration	**45.4%**	**14.1%**	**18.1%**	**6.0%**
Median	**12.6%**	**3.8%**	**3.0%**	**3.8%**

Table 2. **Results: track length**. Average improvement of correct track length in example sequences of the proposed motion model over the three existing models.

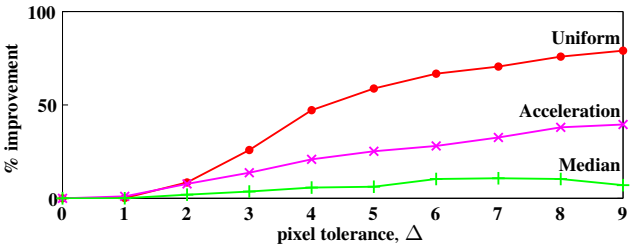

Figure 2. **Results: track length**. Average improvement in track lengths over the existing motion models across all four sequences against the accuracy threshold Δ (deviation from ground-truth). Our proposal outperforms the alternatives at all levels.

all cases, with improvements of up to 80%, 45% and 12% over the uniform, acceleration and median models respectively. Figure 2 shows how the mean track length improvement, averaged over the four sequences, increases roughly linearly with Δ.

A few track case-studies are worth comments. In the giraffe sequence, the motion on the ear is very different to that in the rest of the scene. Even though it is a relatively small object, its motion was successfully captured by the motion model. The mouth video is challenging mainly for the untextured subject. Despite this, the global motion model was able to successfully support motion in the problematic regions. The leopard sequence also provides a difficult set of features to track, having large non-rigid deformations, significant motion blur and very self-similar texture. However, the combination of local and global motion information is able to guide the tracker through these difficulties. The zebras are a good example of the power of the sliding temporal window: methods that try to find a global solution are likely to fail here because of the large number of independent non-rigid objects in the scene. By considering 10-frame sub-sequences, rank-6 motion models were adequate to cover the complex motions observed.

A second experiment investigates the *predictive power* of the proposed motion model, compared to the three alternatives. It was calculated as the RMS pixel error of the predictions made by each model using the ground truth data as the prior observations. The results, presented in Table 3, show that improvements of up to 80%, 50% and 20%, over each of the standard models respectively, are possible. On the giraffe seqence, the median filter predicted slightly better (despite performing less well when the whole system is

Sequence	**Proposed**	Uniform	Acceleration	Median
Giraffe	**1.85**	2.92 (**36.6%**)	2.62 (**29.4%**)	1.67 (**-10.5%**)
Leopard	**1.61**	3.79 (**57.4%**)	2.59 (**37.6%**)	1.83 (**11.9%**)
Mouth	**1.72**	7.89 (**78.2%**)	3.49 (**50.6%**)	1.75 (**1.4%**)
Zebras	**1.22**	2.68 (**54.6%**)	2.03 (**40.1%**)	1.52 (**20.0%**)

Table 3. **Results: predictive power**. The average RMS error, in pixels, of predictions using the ground truth data. Percentage improvement achieved by our model is given in brackets. Here, 'Uniform' is the constant position model and gives a scale reference.

considered), probably due to the high texture density in this scene.

Only a small number of tracks, those for which we have ground truth, have been discussed here, but the qualitative performance is typical. It is important to note that the quality of the tracks were improved in all cases through the use of the motion prior, that is, the use of the extra information very rarely degrades the results and often provides a substantial increase in performance.

The impact on computational speed is primarily in the computation of the motion models, with complexity approximately equivalent to the second-pass stage, so that the addition of priors approximately doubles the computational complexity. For an interactive tracking scenario, the primary speed requirement is in pass two, as the motion models can be precomputed when the sequence is loaded from tape or scanned. In this case, a speed advantage is enjoyed when the prior is tight because the size of the search window is reduced.

5. Concluding Remarks

We have shown that local feature tracking can be guided using global scene motion information in an efficient algorithm. By making predictions of the new location of a feature point using a low rank approximation to the motion in the rest of the frame, tracking algorithms can be made to generate longer and more accurate tracks. The confusion caused to standard tracking algorithms, by repeated texture and ill-defined templates, can be mitigated with this prior.

Some characteristics of the algorithm are worthy of note. Even though the track prediction is a simple linear projection (Eq. 7), it can model complex non-rigid motions and can predict from tracks with jumps and cusps, because of the use of the long time history and because of the fitting of the global model. In effect we have a high-order Kalman filter where the state transition matrix is specialized to every frame of the sequence.

Technically, this "prior" is not independent of the information used in the calculation of the likelihood, both being derived from the pixel intensity values of the image sequence. In practice, though, the distance between the feature being tracked and the points used for the motion model

is large enough, on average, to allow us to treat the two as being independent.

We have used a relatively simple base tracker into which to impose the priors, ignoring much of the recent work on template updating, occlusion, and other non-motion priors [7, 10]. It would appear reasonable, however, to expect that the global motion model will improve any base tracker that uses a simpler (or no) motion model.

A salient issue not addressed here was that of initialization. For the first $\mathtt{M}$ frames of a sequence, when motion predictions can not be made, an alternative strategy must be used. For the examples given in this paper, the prior was set to be uniform for these initial frames, *i.e.* the trackers compared had identical behaviour for those first frames to aid the comparison. More sophisticated methods can easily be employed, such as multiple hypothesis techniques. Once enough frames have been tracked, selection between alternative hypotheses can be aided using the prior for the full tracks, *i.e.* using the motion basis for those frames to calculate reprojection errors and coefficient distances.

Appendix

Here we go through the derivation of Eq. (13), *i.e.* the evaluation of the integral in (12) and the determination of the diffused prior, $\mathrm{q}(\mathbf{x}_t)$. The first steps are manipulations of the integrand in (12), which takes the form $\exp(-\gamma \mathrm{E}(\mathbf{x}, \mathbf{u}))$. For brevity, $\mathbf{x}_t$ will be written simply as '$\underline{\mathbf{x}}$' and $\mathbf{z}_{t-\mathtt{W}}$ will be represented by '$\underline{\mathbf{z}}$'. We start by defining

$$\mathrm{A} = \gamma^{-1}\Sigma_{t-\mathtt{W}}^{-1}, \tag{18}$$

so we can write

$$\begin{aligned}
\mathrm{E} &= \|\underline{\mathbf{x}} - \mathtt{P}\mathbf{u}\|^2 + (\mathbf{u}-\underline{\mathbf{z}})^\top \mathrm{A}(\mathbf{u}-\underline{\mathbf{z}}) && (19)\\
&= \mathbf{u}^\top\mathtt{P}^\top\mathtt{P}\mathbf{u} - 2\underline{\mathbf{x}}^\top\mathtt{P}\mathbf{u} + \underline{\mathbf{x}}^\top\underline{\mathbf{x}} + \mathbf{u}^\top\mathrm{A}\mathbf{u} - 2\underline{\mathbf{z}}^\top\mathrm{A}\underline{\mathbf{z}} + \underline{\mathbf{z}}^\top\underline{\mathbf{z}} && (20)\\
&= \mathbf{u}^\top(\mathtt{P}^\top\mathtt{P}+\mathrm{A})\mathbf{u} - 2\mathbf{u}^\top(\mathtt{P}^\top\underline{\mathbf{x}} + \mathrm{A}^\top\underline{\mathbf{z}}) + \underline{\mathbf{x}}^\top\underline{\mathbf{x}} + \underline{\mathbf{z}}^\top\underline{\mathbf{z}}. && (21)
\end{aligned}$$

Now, with $\mathrm{C} = (\mathtt{P}^\top\mathtt{P}+\mathrm{A})$ and $\mathrm{c} = \mathrm{C}^{-1}(\mathtt{P}^\top\underline{\mathbf{x}} + \mathrm{A}^\top\underline{\mathbf{z}})$:

$$\mathrm{E} = (\mathbf{u}-\mathrm{c})^\top\mathrm{C}(\mathbf{u}-\mathrm{c}) - \mathrm{c}^\top\mathrm{C}\mathrm{c} + \underline{\mathbf{x}}^\top\underline{\mathbf{x}} + \underline{\mathbf{z}}^\top\underline{\mathbf{z}}, \tag{22}$$

allowing the diffused prior to be seen as

$$\mathrm{q}(\underline{\mathbf{x}}) \propto \int e^{-\gamma \mathrm{g}(\mathbf{u})} e^{-\gamma \mathrm{f}(\underline{\mathbf{x}})} e^{-\gamma \underline{\mathbf{z}}^\top\underline{\mathbf{z}}} \mathrm{d}\mathbf{u} \tag{23}$$

i.e. $\mathrm{q}(\underline{\mathbf{x}}) \propto e^{-\gamma \mathrm{f}(\underline{\mathbf{x}})}$, since $\mathbf{u}$ is constant and the $\underline{\mathbf{x}}$s in $\mathrm{g}(\mathbf{u})$ are contained only in the 'mean' term (disappearing on intergration). Examining $\mathrm{f}(\underline{\mathbf{x}})$,

$$\begin{aligned}
\mathrm{f}(\underline{\mathbf{x}}) &= \underline{\mathbf{x}}^\top\underline{\mathbf{x}} - \mathrm{c}^\top\mathrm{C}\mathrm{c} && (24)\\
&= \underline{\mathbf{x}}^\top\underline{\mathbf{x}} - (\mathtt{P}^\top\underline{\mathbf{x}} + \mathrm{A}^\top\underline{\mathbf{z}})^\top\mathrm{C}^{-1}\mathrm{C}\mathrm{C}^{-1}(\mathtt{P}^\top\underline{\mathbf{x}} + \mathrm{A}^\top\underline{\mathbf{z}}) && (25)\\
&= \underline{\mathbf{x}}^\top\underline{\mathbf{x}} - (\underline{\mathbf{x}}^\top\mathtt{P} + \underline{\mathbf{z}}^\top\mathrm{A})(\mathtt{P}^\top\mathtt{P}+\mathrm{A})^{-1}(\mathtt{P}^\top\underline{\mathbf{x}} + \mathrm{A}^\top\underline{\mathbf{z}}) && (26)\\
&= \underline{\mathbf{x}}^\top(\mathtt{I} - \mathtt{P}(\mathtt{P}^\top\mathtt{P} + \mathrm{A})^{-1}\mathtt{P}^\top)\underline{\mathbf{x}} + \\
&\qquad -2\underline{\mathbf{x}}^\top\mathtt{P}(\mathtt{P}^\top\mathtt{P} + \mathrm{A})^{-1}\mathrm{A}^\top\underline{\mathbf{z}} + \ldots && (27)
\end{aligned}$$

The next stage follows from equating $e^{-\gamma \mathrm{f}(\underline{\mathbf{x}})}$, and hence $\mathrm{q}(\underline{\mathbf{x}})$, with a Gaussian, $\exp(-(\underline{\mathbf{x}}-\mu)^\top\Sigma^{-1}(\underline{\mathbf{x}}-\mu))$ and from there determining the Gaussian's parameters:

$$\begin{aligned}
\gamma\mathrm{f}(\underline{\mathbf{x}}) &\equiv \underline{\mathbf{x}}^\top\Sigma^{-1}\underline{\mathbf{x}} - 2\underline{\mathbf{x}}^\top\Sigma^{-1}\mu + \ldots && (28)\\
\therefore\quad \Sigma^{-1} &= \gamma(\mathtt{I} - \mathtt{P}(\mathtt{P}^\top\mathtt{P} + \mathrm{A})^{-1}\mathtt{P}^\top) && (29)\\
&= \gamma(\mathtt{I} + \mathtt{P}\mathrm{A}^{-1}\mathtt{P}^\top)^{-1} && (30)\\
&= (\gamma^{-1}\mathtt{I} + \mathtt{P}\Sigma_{t-\mathtt{W}}\mathtt{P}^\top)^{-1} && (31)\\
\text{and}\quad \Sigma^{-1}\mu &= \gamma\mathtt{P}(\mathtt{P}^\top\mathtt{P} + \mathrm{A})^{-1}\mathrm{A}^\top\underline{\mathbf{z}} && (32)\\
&= \gamma(\mathtt{I} - \mathtt{P}(\mathtt{P}^\top\mathtt{P} + \mathrm{A})^{-1}\mathtt{P}^\top)\mathtt{P}\underline{\mathbf{z}} && (33)\\
&= \Sigma^{-1}\mathtt{P}\underline{\mathbf{z}} && (34)
\end{aligned}$$

where (30) is an application of the Sherman–Morrison–Woodbury identity[2] and (33) employs $\mathrm{A} = \mathrm{A}^\top$ plus $(\mathrm{B}+\mathrm{A})^{-1}\mathrm{A} = \mathtt{I} - (\mathrm{B}+\mathrm{A})^{-1}\mathrm{B}$ and the Newton's Cradle identity[3]. Equation (13) follows.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Trans. Sig. Proc.*, 50(2):174–188, 2002.

[2] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proc. ECCV*, pages 683–695, 1996.

[3] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proc. CVPR*, volume 2, pages 690–696, 2000.

[4] G. J. Brostow and R. Cipolla. Unsupervised Bayesian detection of independent motion in crowds. In *Proc. CVPR*, volume 1, pages 594–601, 2006.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cam. Uni. Press, 2^{nd} ed., 2004.

[6] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Proc. ICCV*, 1999.

[7] A. Jepson, D. Fleet, , and T. El-Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10), 2003.

[8] X. Llado, A. Del Bue, and L. Agapito. Non-rigid 3D factorization for projective reconstruction. In *Proc. BMVC*, 2005.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. Comp. Vis.*, 60(2):91–110, 2004.

[10] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. In *Proc. BMVC*, 2003.

[11] P. Sand and S. Teller. Video matching. *ACM Trans. Graph.*, 23(3):592–599, 2004.

[12] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. In *Proc. CVPR*, 2006.

[13] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. In *Proc. ECCV*, 2004.

[14] P. Smith, D. Sinclair, R. Cipolla, and K. Wood. Effective corner matching. In *Proc. BMVC*, 1998.

[15] L. Torresani and A. Hertzmann. Automatic non-rigid 3D modeling from video. In *Proc. ECCV*, 2004.

[2] Def: $(\mathtt{A}+\mathtt{UKV})^{-1} = \mathtt{A}^{-1} - \mathtt{A}^{-1}\mathtt{U}(\mathtt{K}^{-1}+\mathtt{V}\mathtt{A}^{-1}\mathtt{U})^{-1}\mathtt{V}\mathtt{A}^{-1}$

[3] Def: $\mathtt{V}(\mathtt{I}+\mathtt{U}\mathtt{V}) = (\mathtt{I}+\mathtt{V}\mathtt{U})\mathtt{V}$.

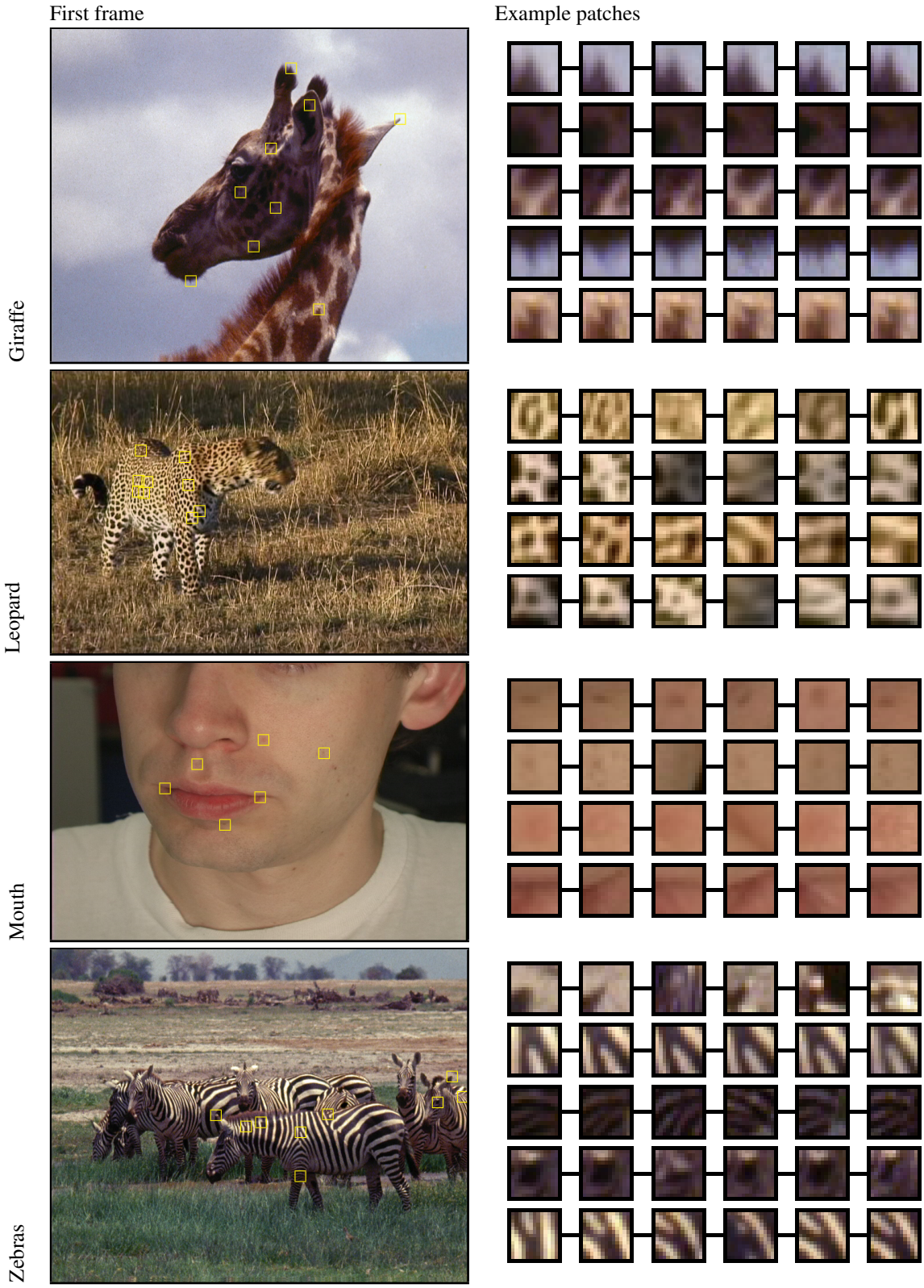

Figure 3. The first frame of each sequence used in the evaluation, with the start of the ground truth tracks shown. On the right is an indication of the range of appearance variation in the feature tracks used. Note that feature choice was limited to features that a) could be tracked consistently and accurately by a human and b) were continuously visible for the whole sequence. See §4 for more comments.