Motion Models that Only Work Sometimes

Cristina Garcia Cifuentes, Marc Sturzel, Frédéric Jurie, Gabriel J. Brostow

Motion Models that Only Work Sometimes

Cristina García Cifuentes[1]
c.garciacifuentes@cs.ucl.ac.uk
Marc Sturzel[3]
marc.sturzel@eads.net
Frédéric Jurie[2]
https://users.info.unicaen.fr/~jurie/
Gabriel J. Brostow[1]
http://www.cs.ucl.ac.uk/staff/g.brostow

[1] University College London, UK
[2] University of Caen, France
[3] EADS Innovation Works, France

Abstract

It is too often that tracking algorithms lose track of interest points in image sequences. This persistent problem is difficult because the pixels around an interest point change in appearance or move in unpredictable ways. In this paper we explore how classifying videos into categories of camera motion improves the tracking of interest points, by selecting the right specialist motion model for each video. As a proof of concept, we enumerate a small set of simple categories of camera motion and implement their corresponding specialized motion models. We evaluate the strategy of predicting the most appropriate motion model for each test sequence. Within the framework of a standard Bayesian tracking formulation, we compare this strategy to two standard motion models. Our tests on challenging real-world sequences show a significant improvement in tracking robustness, achieved with different kinds of supervision at training time.

frame 0 frame 10 frame 20

Figure 1: Example sequence with overlaid box showing the output of our specialized "forward" motion model, where the velocity and scale of objects approaching the camera tend to increase. Neither Brownian nor constant-velocity motion models are as successful at tracking interest points here.

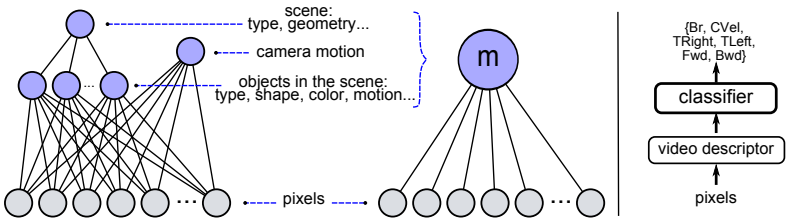


Figure 2: (Left) Illustration of the complex combination of causes that produce motion in a video. (Middle) Approximating the real graphical model, the many causes, which may even have semantic meanings, can be grouped into one more abstract variable. (Right) In practice, we quantize the causes into a discrete variable, and infer it using a discriminative model.

1 Introduction

The 2D motion of objects in a video is strongly conditioned by the camera motion, as well as the type of scene and its geometry. Each object may have its own motion pattern in the context of the more global scene-dependent motion (see Figure 1). We are interested in motion models that help with overall tracking of interest points. As shown in the supplementary material, Brownian and constant-velocity motion models are often ineffective, despite having a reputation as general-purpose models. We propose that conditioning the choice of motion model on the "causes" of motion (see Figure 2) can improve tracking performance because the best available specialist motion model can be employed for each video. Such conditioning frees the specialist from having to cope with motions beyond its capabilities. In our prototype, the specialist motion models are programmed in advance by hand, but they could be created by other means in principle.

Given a set of specialist motion models, we must then automatically determine which model to apply when tracking interest points in a given sequence. This choice among models amounts to video categorization. We categorize by means of supervised classification.

Our main contributions are **(i)** the design of a few specialized motion models, which only track well in scenes having similar geometry and camera motion, **(ii)** showing that a new video can be mapped to one of the available motion models automatically, **(iii)** a new dataset to evaluate 2D-point tracking, and critically **(iv)** experimental evidence of improved tracking performance when using the predicted motion model in unseen videos.

2 Related work

We track a moving target as a dynamic system. Online trackers represent the current situation by updating a state vector. Trackers can be seen as having two main components [1]: a model describing the evolution of the state with time, *i.e.* the *dynamic model*, and a model relating the noisy observations to the state, *i.e.* the *measurement model*. We focus on choosing the dynamic model that is most appropriate for each video clip, though other related tracking research is summarized here as well.

Measurement models. The measurement model is often related to visual appearance cues, such as contours [8] or color [21]. Online adaptation of the measurement model is an ongoing area of study, *e.g.* [9, 32], the main issue being the trade-off between adaptability versus drift. A time-weighted appearance model can be obtained through weighted online

learning [33] that avoids drift and emphasizes most recent observations. When the tracked object is at least partially known, an appearance model can be learned *a priori*, or revised online, such as Grabner *et al*.'s online boosting tracker [7]. Tracking-by-detection uses temporal constraints to resolve the space of hypotheses that emerge when flexible appearance models are used. Kalal *et al*. [10] have an impressive example of a real-time system for solitary patches, while Prisacariu and Reid [22] demonstrate excellent tracking of even articulated shapes using level sets and nonlinear shape manifolds as shape priors for known object classes. Williams *et al*. [31] train a regressor which can localize the tracked target even with significant occlusions.

An appearance descriptor can be enhanced by integrating simple motion information. For example, Kristan *et al*. [14] use a measurement model combining color and optical flow to resolve color-only ambiguities. They use a mixed dynamic model combining constant velocity for position and Brownian evolution for size. Čehovin *et al*. [28] built on that approach with a constellation of local visual parts, which get resampled spatially based on consistency with an object's higher-level appearance. Most relevant to our supervised learning-based approach, Stenger *et al*. [27] assess the suitability of different measurement models for a particular tracking scenario, *e.g*. face or hand tracking. Their experiments focused on coupling different models of appearance, so they replace the dynamic model by simple exhaustive search, and re-initialize lost tracks as needed. In contrast, we use a fixed appearance model to better study the impact of choosing different motion models.

Dynamic models. Comparatively little attention has been paid to motion models. Dynamic models are important because they allow the search to be guided, tracking through short periods of occlusion and favoring observations with more likely motion. While it is very common to simply use Brownian or constant-velocity motion models, with parameters set *a priori*, their generality hurts them in many specific situations. Lourenço and Barreto [17] show the importance of an appropriate motion model in the case of scenes with radial distortion. If tracking an object with a known nature, a physical model can better dictate the system's predictions [5]. If ground-truth tracks are available, Blake *et al*. [2] showed that it is possible to learn the parameters of a motion model for a given scenario. For simple cases, the initial tracks can come from an untrained tracker. Rodriguez *et al*. [23] use topic models to learn long-term flow patterns from a sequence of a crowded scene, which improves re-tracking in the same scene. In posterior work [24] they also cope with previously unseen sequences. For a given area, they retrieve training patches that appear similar. They use the learned flow patterns as a prior in the framework of a Kalman filter [11] with a constant-velocity motion model and a flow-based measurement model. Buchanan and Fitzgibbon [3] obtain a robust motion prior from the global motion of 2D points in a time window, and apply it to re-track them. That motion model is based on rank constraints on the matrices explaining all the initial 2D tracks, without being restricted to rigid motions. While their model focused on translations of the individual points, more complex transformations are possible.

Outgrowing the closed-form learning that was possible in [2], North *et al*. [20] present an expanded algorithm to learn the parameters of more complex dynamics. Illustrated on the example of juggling, they model motion as a sequence of motion "classes", and concurrently learn the parameters of each motion class and the transition probabilities among classes. The large variety of both motion models and available appearance models makes it difficult to choose among them when designing a system. Instead of a premature and hard commitment for any particular combination, Kwon and Lee [15] adaptively sample from a tracker space integrating appearance and motion models, selecting the tracker that provides the highest

data likelihood. We approach a similar challenge, but choosing among very specialized motion models, and leveraging supervised learning.

Kowdle and Chen [13] also train classifiers based on the dynamism of both scene and camera, and use them together with shot-boundary detection in order to segment videos into clips of consistent motion. From the model-selection point of view, our work is related to Mac Aodha *et al.*'s [18]. They train a supervised classifier to predict the per-pixel success of different flow algorithms, thereby allowing for comparisons among multiple competing flow-estimates.

3 Dynamic models and choosing among them

We focus on four types of camera motion (traveling right, traveling left, moving forward, moving backward) and implement specialized dynamic models for each. They are described below, together with two standard dynamic models used very frequently for tracking, namely Brownian motion and constant velocity. We establish six video categories that correspond to the six dynamic models, and aim to categorize each new video sequence to its most appropriate dynamic model. These coarse categories are simple enough to be automatically recognized much of the time, yet emerge as specific enough to give improved performance when tracking with the corresponding dynamic model. To validate our approach, a dataset (Section 3.2) of challenging videos was collected and augmented with manual annotations of correct tracks. Section 3.3 describes the video-clip classification process.

3.1 Dynamic models

To compare the performance of the dynamic models on any given video, we choose the task of tracking 2D points. We adopt the standard Bayesian formulation and implement it as a particle filter [1] with a fixed measurement model. Given the position of a 2D point in an initial frame, we want to know the position of that point in the subsequent frames. For each frame t, we estimate the posterior distribution of the state of the system $\mathbf{x}_t$ given the observed images $\mathbf{I}_{1..t}$ up to time t,

$$\underbrace{p(\mathbf{x}_t|\mathbf{I}_{1:t})}_{\text{posterior}} = \underbrace{p(\mathbf{I}_t|\mathbf{x}_t)}_{\text{measurement model}} \int \underbrace{p(\mathbf{x}_t|\mathbf{x}_{t-1})}_{\text{dynamic model}} \underbrace{p(\mathbf{x}_{t-1}|\mathbf{I}_{1..t-1})}_{\text{prior}} d\mathbf{x}_{t-1}, \tag{1}$$

where the posterior at time $(t-1)$ becomes the prior at time t. We now present the models in terms of equations describing the evolution of the system given the previous state. Implementation details can be found in Section 4.1 and the supplementary material.

Brownian (Br): This is the baseline dynamic model. The system's state vector is $\mathbf{x} = [x,\ y,\ w,\ h]$, where $[x,y]$ is the 2D position and $[w,h]$ are the patch width and height. Position and scale change only due to Gaussian noise. We keep the aspect ratio fixed, so

$$[x_{t+1},\ y_{t+1}] \sim \text{Norm}\left([x_t,\ y_t], \text{diag}(\sigma_x^2, \sigma_y^2)\right), \tag{2}$$

$$[w_{t+1},\ h_{t+1}] = s[w_t,\ h_t]; \quad s \sim \text{Norm}(1, \sigma_s^2). \tag{3}$$

Constant Velocity (CVel): This model is also a standard choice for tracking. We set $\mathbf{x} = [x,\ y,\ w,\ h,\ \dot{x},\ \dot{y},\ s]$, where $[\dot{x},\dot{y}]$ are the displacements relative to the previous time-step, and

s is the last increase in scale. The new displacement is similar to the previous one, *i.e.* nearly constant velocity. Scale is assumed to be Brownian, so

$$[\dot{x}_{t+1},\, \dot{y}_{t+1}] \sim \text{Norm}\left([\dot{x}_t,\, \dot{y}_t], \text{diag}(\sigma_x^2, \sigma_y^2)\right), \tag{4}$$

$$[x_{t+1},\, y_{t+1}] = [x_t,\, y_t] + [\dot{x}_{t+1},\, \dot{y}_{t+1}], \tag{5}$$

$$[w_{t+1},\, h_{t+1}] = s_{t+1}[w_t,\, h_t]; \quad s_{t+1} \sim \text{Norm}(s_t, \sigma_s^2). \tag{6}$$

In our experiments we set $\sigma_y = \sigma_x$.

Traveling Right / Left (TRight / TLeft): These models are specialized to scenes in which the camera moves horizontally to the right (or left). The model can be written with the same equations as the constant-velocity model, only $\sigma_y << \sigma_x$. The prior on the x displacement in TRight has the opposite sign to TLeft.

Forward / Backward (Fwd / Bwd): When the camera moves horizontally forward in a scene, some objects seem to appear at a 2D focus of expansion point $[f^x, f^y]$, and move away from it and grow as they get closer to the camera. The opposite happens when the camera moves backward. Based on validation data, we designed a dynamic model specialized for such scenes. The model assumes that the distance between a tracked point and its focus of expansion increases (or decreases) multiplied by a nearly-constant factor d. The patch size also has a nearly-constant increasing rate s. Both rates are coupled. Fwd and Bwd models have opposite priors on these rates ($d, s >= 1$ for Fwd and $d, s <= 1$ for Bwd). The state vector is $\mathbf{x} = [x,\, y,\, w,\, h,\, d,\, s,\, f^x,\, f^y]$.

We model each 2D point as having its own focus of expansion (which is sampled from the one in the previous time-step) and moving from it in a nearly-straight line passing through the current position. We account for slight deviations through the variable α, which represents a tilt angle whose corresponding rotation matrix is $\mathbf{R}_\alpha$. The state updates are

$$[f^x_{t+1},\, f^y_{t+1}] \sim \text{Norm}([f^x_t,\, f^y_t], \sigma_f^2 \mathbf{I}), \tag{7}$$

$$d_{t+1} = d_t + \sigma_d \varepsilon_1; \quad \varepsilon_1 \sim \text{Norm}(0, 1.0), \tag{8}$$

$$s_{t+1} = s_t + \sigma_s \varepsilon_2; \quad \varepsilon_2 \sim \text{Norm}(\varepsilon_1, 0.1), \tag{9}$$

$$\alpha \sim \text{Norm}(0, \sigma_\alpha^2), \tag{10}$$

$$[x_{t+1},\, y_{t+1}]^T = d_{t+1} \mathbf{R}_\alpha \begin{bmatrix} x_t - f^x_{t+1} \\ y_t - f^y_{t+1} \end{bmatrix} + \begin{bmatrix} f^x_{t+1} \\ f^y_{t+1} \end{bmatrix}, \tag{11}$$

$$[w_{t+1},\, h_{t+1}] = s_{t+1}[w_t,\, h_t]. \tag{12}$$

Note that Brownian motion is a particular case of these two models, when the rates controlling scale and distance-to- focus are equal to one. Besides, there might be ambiguity between the Fwd and Bwd models: the same rectilinear motion can be explained as approaching a point in the direction of the motion or as moving away form a point placed in the opposite direction. This means that both models can equally explain the observed motion of an object if the changes in scale are subtle. When the changes become more obvious, the wrong combination of focus position and motion direction will loose track of the object.

3.2 Dataset

While datasets of manually tracked objects exist (*e.g.* [29]), we needed self-consistent video clips spanning a variety of camera motions, and ground-truth 2D tracks for points rather

than objects. Our dataset consists of over 100 challenging real-world videos from YouTube, encompassing various scenarios, illuminations, and recording conditions. They last typically between three and 90 seconds. We hired users to use our annotation tool to pick at least 10 interest points per video, among all FAST [25, 26] interest points. They marked each point's $[x, y]$ position in at least 10 subsequent frames, sparsely in time if possible. These annotations serve for assessing tracking performance, as described in Section 4.2.

To train a supervised classifier, we manually obtain *inspection-based* labels for each video. We ourselves inspected the videos and judged whether a clip belonged predominantly to one of the four camera motions or might be better served by the Br or CVel models. For the TRight and TLeft categories, we also include the flipped video in the dataset. In total there are 12 videos labeled as Br, 11 as CVel, 11 (+ 17 flips) as TRight, 17 (+ 11) as TLeft, 24 as Fwd, and 14 as Bwd. The flipped videos provide some symmetry with regards to the difficulty of tracking-left and tracking-right videos. One might think that this introduces redundancy in the dataset and makes the classification task easier. In fact, such videos look similar but have different target labels: the similarity in appearance becomes a handicap in this classification task.

An alternative way of obtaining supervision is *performance-based* labeling. In this case, for a given set of available motion models and parameter settings, a video is labeled with the one that performed best according to the annotations of ground-truth tracks.

Note that the specialized motion models have been designed previous to obtaining this benchmark data, and in particular, previous to establishing the *inspection-based* (and potentially subjective) target categories. We use default parameters in our dynamic models and no tuning has been done for this data.

3.3 Video description and classification

We use the local video features presented by Wang *et al*. [30] for the task of action recognition. They sample points densely and obtain 15-frame-long tracks by concatenating median-filtered optical flow. From the local (x, y, t)-tube around each track, they extract four descriptors: the trajectory itself (Traj), a histogram of oriented gradients (HOG), a histogram of oriented flow (HOF), and histograms of flow gradients called motion-boundary histograms (MBH). We quantize these local features using Extremely Randomized Trees [8] as in [19] and obtain visual-word vocabularies. For each type of descriptor, we train five trees of around 1000 leaves each. For the experiments, rather than computing the vocabulary on our dataset, we trained on completely different data, namely the action recognition YouTube dataset [16]. We obtain global video signatures by computing the normalized histogram of visual words, one per type of descriptor.

For video classification, we use an existing multiclass SVM [4]. It consists of an all-pairs biclass SVM and majority voting. We pre-compute χ^2-kernels for each type of signature, and combine different types of signatures through the sum of the kernels. This approach has been tested on short clips in which a constant motion model is assumed, but one could easily imagine a sliding-window extension to deal with longer, more varied videos.

4 Experiments and results

This section experimentally validates our approach, showing that we can classify a video clip to select among motion models and significantly improve tracking performance. We outline

the tracking implementation and how the performance of point trackers is evaluated. We then show the tracking performance of the six dynamic models, measuring how each performs on its own. We finally report the performance of the video classification algorithm and, critically, the results of point trackers using motion models predicted by the video classifier.

4.1 Tracking implementation

We implement (1) as a particle filter. Our motion models have higher-dimensional state vectors than the baselines and therefore might enjoy a larger number of particles, but we fix it to 50 for all the models so that the comparisons are fair in terms of allocated resources.

The state vector $\mathbf{x}_t$ contains at least the 2D position of the point $[x_t, y_t]$ and the dimensions $[w_t, h_t]$ of a patch around it. We compare trackers' performance regarding only the position estimates that they produce, but width and height are also needed by the measurement model.

The measurement model estimates the likelihood of the observed image given a state. Given the initial position of the 2D point $[x_1, y_1]$ in the initial frame $\mathbf{I}_1$, we extract a 15×15-pixel patch around it to keep as the reference template.

The likelihood of any state in any subsequent frame t is measured by first extracting a patch from image $\mathbf{I}_t$ at the prescribed position and scale. Once interpolated, the patch is scored based on normalized cross-correlation (NCC) to the reference template,

$$p(\mathbf{I}_t|\mathbf{x}_t) \propto f(NCC(Patch_{ref}, Patch_t)). \tag{13}$$

The mapping $f(\cdot)$ aims at being a calibration, transforming raw NCC values to be proportional to likelihood values. $f(\cdot)$ is adapted to each tracked point, and is computed in the first frame by measuring the NCC between the template patch and a number of neighboring patches. We force the mapping to be monotonic and compute it efficiently as detailed in the supplementary material.

4.2 Performance evaluation

We evaluate the trackers' performance in terms of robustness, on the dataset in Section 3.2.

Let $p_i = [x_i, y_i]$, $i \in [1, \ldots, N]$ be the position of the N ground-truth points in a given video. Let $\hat{p}_i$, be the position of the reference point p_i estimated by the tracker. We compute the percentage of successful track estimates, *i.e.* the fraction of point-positions that match the ground-truth locations within a certain level of precision, so

$$robustness = \frac{1}{N}\sum_{i=1}^{N} \triangle_\theta(p_i, \hat{p}_i); \quad \triangle_\theta(p_i, \hat{p}_i) = \begin{cases} 1 & \text{if } |x_i - \hat{x}_i| < \theta \text{ and } |y_i - \hat{y}_i| < \theta \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

This measure of performance is averaged over two runs of tracking. The trackers are not re-initialized after loss of track, so early loss of track is typically more costly. To obtain the global performance of the whole dataset, we average the robustness obtained when using the motion model that was selected for individual videos. In all our experiments we set θ to 10. This is a quite tolerant threshold given the initial patch size, and therefore allows to distinguish inaccurate tracking from complete loss. Other values could be used.

4.3 Tracking robustness of individual motion models

The performance of the six motion models is computed over the whole dataset. As shown in Table 1, no individual model performs extraordinarily overall. The hypothesis that ideally, each video should be processed using the most appropriate motion model is explored through the following tests.

We estimate the *idealized* performance that is obtained if, among the available motion models, an oracle chose the one that actually performs best on each video. The ideal performance is already better when there is a choice between just the two standard motion models (0.493). Enlarging the collection with our four specialized models raises the ceiling further (0.561). Knowing the rather subjective inspection-based labels yields a best-case score of 0.526, which is worse than the six-way performance-based oracle, but a significant improvement nonetheless.

Figure 3 gives insight into the behavior of the different motion models. It shows the tracking robustness obtained on each group of videos with each motion model, for different parameter settings. The videos are grouped according to their inspection-based labels, and the performance has been averaged within groups. One can see the inclination of most groups toward their corresponding dynamic model. For example, TRight and TLeft are poor in general, but they largely outperform the rest on their corresponding families of videos. On videos labeled as Constant Velocity, CVel outperforms the rest of models in the case of appropriate parameter settings. Between the two baselines, CVel tends to outperform Br, but there is no clear winner as it depends of the parameter settings. Br performs better than CVel on videos labeled as Brownian. Fwd and Bwd perform better than Br in general, except for videos labeled as Brownian, on which they are nevertheless more robust when the level of noise in the models is high. We observe also that Fwd and Bwd perform very similarly, as TRight and TLeft, which is explained by the fact that they only differ in the prior.

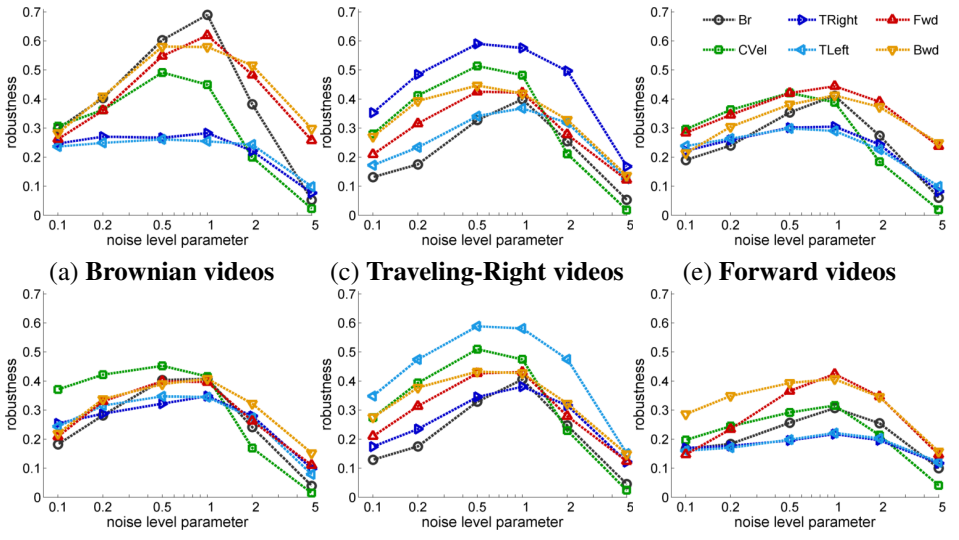
	Individual motion models						Ideal predictions			Our method
	Br	CVel	TRight	TLeft	Fwd	Bwd	best{Br, CVel}	best{all}	manual labels	
tracking robustness ($\cdot 10^{-2}$)	42.3	43.2	37.9	37.2	44.7	43.7	49.3	56.1	52.6	51.9
$\pm$ std. dev. random runs	0.4	0.4	0.7	0.5	0.2	0.1	0.6	0.4	0.2	0.1
% best choice (± 2)	21	11	12	20	20	16	32	100	52	50

Table 1: White background: average tracking robustness of each individual motion model over all videos (default parameters). Bottom row: percentage of times *that* motion model (among six) was the best choice. Light gray: best-case results if model is selected by either a performance-based oracle, or our inspection-based labels. Right column: tracking each video using our classifier's suggested motion model, using inspection-based training data.

4.4 Tracking robustness using classification

The previous experiments showed how predicting the most appropriate motion model would lead to significant improvement of tracking robustness. We propose that such predictions can be made using a set of training videos, assigning them to categories that encapsulate shared motion properties, and using a classifier to predict the category of a new video at test time.

We train multiclass SVMs using different video descriptors (see Section 3.3) and the *inspection-based* category labels, then evaluate them in a leave-one-out manner so that each



(a) **Brownian videos** (c) **Traveling-Right videos** (e) **Forward videos**

(b) **Constant-Velocity videos** (d) **Traveling-Left videos** (f) **Backward videos**

Figure 3: Average tracking robustness of each motion model (see legend) on each inspection-labeled group of videos. Different parameter settings, from low to high values of noise parameters σ_x, σ_y, σ_s, σ_d, etc. are controlled by a single noise level parameter (default = 1). Considering peak-performance and area under the curves, groups are partial to their respective motion models. Motion models differ in their robustness to parameter settings.

	HOG	Traj	HOF	MBH	Traj + HOF + MBH	Target labels
classification accuracy (%)	30	84	85	83	91	–
tracking robustness ($\cdot 10^{-2}$)	42.2	52.2	52.09	50.5	51.9	52.6
$\pm$ std. dev. random runs	0.6	0.1	0.01	0.4	0.1	0.2

Table 2: Classifiers trained using different descriptors. First row: classification accuracy with respect to the target inspection-based labels (leave-one-out training). Below: tracking robustness obtained using the model predicted for each sequence. Note how features other than HOG lead to tracking robustness that is superior to the general-purpose motion models.

video in the dataset acts as the test video once. This produces a motion category prediction for each video. Table 2 shows the classification performance of the different video descriptors with regard to the target inspection-based labels. The table also shows the tracking robustness obtained when tracking with the predicted motion model.

Ignoring HOG, the combination of three motion-related descriptors yields the best classification accuracy. However, this does not translate into the best tracking performance. Not all the classification mistakes are equally costly in terms of tracking. Classification, by design, is optimized here for mutually-exclusive classes and a binary 0-1 loss, *i.e.* the five possible types of error incur the same penalty. In practice, confusing TRight for TLeft could hurt tracking much more than picking CVel, for example.

Performance-based categories The inspection-based labels are subjective, so in the same

manner, we also trained a classifier on performance-based labels. Classification accuracies are much poorer (43% for six categories), indicating that this is a harder classification task than the one targeting inspection-based labels. Nevertheless, the tracking performance is comparable (0.529), which can be explained by the fact that the "best{all}" ideal predictions in Table 1 set a higher ceiling. Note that the human effort to obtain the inspection-based labels is much lower, as no manual ground-truth point tracks are needed, only visual inspection and some knowledge of the models.

5 Discussion

We have presented a simple approach to choose among specialized dynamic models. Pre-classifying each video leads to improved tracking robustness, requiring training-time data and only weak supervision in the form of inspection-based labels. Brownian motion and constant-velocity models have comparable performance in our experiments, while guidance from our inspection- and performance-based classifiers improves this baseline by over 20%. While one can imagine other optimization methods to select among dynamic models, we have so far not found one as effective as the proposed classification approach. In particular, we experimented with a heuristic based on track lengths, which scored half way between the general-purpose baseline models and our method (more details in the supplementary material). Further, classification has the potential to scale well when the number of motion models increases, making the running of each model in a try-and-see approach more costly than extracting and classifying a feature vector.

We have used a particle filter and a simple measurement model only as a platform for comparing dynamic models. Classifying a video according to its dynamic model could be used in combination with (and should be complimentary to) any tracking approach and more sophisticated measurement models.

Several questions arise that could be the object of future work. One unaddressed issue in this work is the automatization of the discovery of motion classes within the training set and the design of adapted models. A first step in this direction would be related to Kitani *et al.*'s algorithm [12] to cluster videos into categories of first-person actions in a fast an unsupervised way. If many dynamic models could be designed systematically, then videos could be mapped to both motion models *and* bespoke parameters. It could also prove useful to apply our method to local regions or in a sliding-window fashion. In addition, the experiments using performance-based supervision highlight that considering the categories as mutually exclusive may be unnecessarily distracting to the classifier. One approach could be to employ regression or structured output spaces to predict the relative success of the motion models. Overall, there is good reason to expect that further very specialized motion models, that are not good in general but outstanding under known circumstances, are worth developing.

References

[1] M. Sanjeev Arulampalam, Simon Maskell, and Neil Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[2] Andrew Blake, Michael Isard, and David Reynard. Learning to track the visual motion of contours. *Artificial Intelligence*, 78:101–134, 1995.

[3] Aeron Buchanan and Andrew W. Fitzgibbon. Combining local and global motion models for feature point tracking. In *CVPR*, 2007.

[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[5] Anne Cuzol and Étienne Mémin. A stochastic filter for fluid motion tracking. In *ICCV*, pages 396–402, 2005.

[6] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[7] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, pages 234–247, 2008.

[8] Michael Isard and Andrew Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[9] Allan D. Jepson, David J. Fleet, and Thomas F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1296–1311, 2003.

[10] Z. Kalal, J. Matas, and K. Mikolajczyk. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[11] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[12] Kris M. Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR*, pages 3241–3248, 2011.

[13] Adarsh Kowdle and Tsuhan Chen. Learning to segment a video to clips based on scene and camera motion. In *ECCV*, 2012.

[14] M. Kristan, J. Perš, S. Kovačič, and A. Leonardis. A local-motion-based probabilistic model for visual tracking. *Pattern Recognition*, 42(9):2160–2168, 2009.

[15] Junseok Kwon and Kyoung Mu Lee. Tracking by sampling trackers. In *ICCV*, pages 1195–1202, 2011.

[16] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, pages 1996–2003, 2009.

[17] Miguel Lourenco and Joao Pedro Barreto. Tracking feature points in uncalibrated images with radial distortion. In *ECCV*, 2012.

[18] Oisin Mac Aodha, Gabriel J. Brostow, and Marc Pollefeys. Segmenting video into classes of algorithm-suitability. In *CVPR*, 2010.

[19] Frank Moosmann, Eric Nowak, and Frédéric Jurie. Randomized clustering forests for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1632–1646, 2008.

[20] Ben North, Andrew Blake, Michael Isard, and Jens Rittscher. Learning and classification of complex dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000.

[21] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In *ECCV*, pages 661–675, 2002.

[22] Victor A. Prisacariu and Ian Reid. Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *CVPR*, pages 2185 –2192, 2011.

[23] Mikel Rodriguez, Saad Ali, and Takeo Kanade. Tracking in unstructured crowded scenes. In *ICCV*, pages 1389–1396, 2009.

[24] Mikel Rodriguez, Josef Sivic, Ivan Laptev, and Jean-Yves Audibert. Data-driven crowd analysis in videos. In *ICCV*, pages 1235–1242, 2011.

[25] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *ICCV*, volume 2, pages 1508–1511, 2005.

[26] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*, pages 430–443, 2006.

[27] Björn Stenger, Thomas Woodley, and Roberto Cipolla. Learning to track with multiple observers. In *CVPR*, pages 2647–2654, 2009.

[28] Luka Čehovin, Matej Kristan, and Ales Leonardis. An adaptive coupled-layer visual model for robust visual tracking. In *ICCV*, pages 1363–1370, 2011.

[29] Carl Vondrick and Deva Ramanan. Video Annotation and Tracking with Active Learning. In *Neural Information Processing Systems (NIPS)*, 2011.

[30] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[31] Oliver Williams, Andrew Blake, and Roberto Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 2005.

[32] Thomas Woodley, Bjorn Stenger, and Roberto Cipolla. Tracking using online feature selection and a local generative model. In *BMVC*, 2007.

[33] Rui Yao, Qinfeng Shi, Chunhua Shen, Yanning Zhang, and Anton van den Hengel. Robust tracking with weighted online structured learning. In *ECCV*, 2012.