A PREDICTOR OF MOVING OBJECTS FOR FIRST-PERSON VISION

Ricardo Sanchez-Matilla and Andrea Cavallaro

Centre for Intelligent Sensing, Queen Mary University of London, UK
{ricardo.sanchezmatilla, a.cavallaro}@qmul.ac.uk

ABSTRACT

Predicting the motion of objects despite the presence of camera motion is important for first-person vision tasks. In this paper, we present an accurate model to forecast the location of moving objects by disentangling global and object motion without the need of camera calibration information or planarity assumptions. The proposed predictor uses past observations to model online the motion of objects by selectively tracking a spatially balanced distribution of keypoints and estimating scene transformations between frame pairs. We show that we can forecast up to 60% more accurately than state-of-the-art alternatives while being resilient to noisy observations. Moreover, the proposed predictor is robust to frame rate reductions and outperforms alternative approaches while processing only 33% of frames when the camera moves. We also show the benefit of integrating the proposed predictor in a multi-object tracker.

Index Terms— Motion model; Prediction model; Moving cameras

1. INTRODUCTION

Motion predictors model the dynamics of objects to estimate, on the image plane or in 3D, their future location. Predictors are designed for static [1, 2] or moving cameras [3, 4], may need camera calibration [5, 6] or to assume that objects move on a common ground plane [7]. Table 1 compares predictors and groups them in two main classes, namely data driven or model based.

Data-driven predictors learn patterns from (large amounts of) training data using machine learning [1, 2, 8, 9]. A clustering-based method can be used to predict the motion pattern of a person in a structured environment [1]. A long short-term memory (LSTM) can predict the location and scale of objects [9], whereas multiple LSTMs can be used to learn the object-to-object interactions to predict the future location of people [2, 8].

Model-based predictors [10, 11, 12] have either no assumptions on object motion [13] or assume that objects maintain a certain velocity learned over recent observations [14, 15, 16]. First or higher order Markov models can be used to model motion [10, 16] and an independent noise component can account for velocity variations (i.e. accelerations) [14, 15]. Acceleration and noise can be modelled based on camera-object distance [17, 18], frame rate [17, 18], or physics [5]. Moreover, non-linear motion patterns can be learned online with a hierarchical association of tracklets [11].

The above predictors are applicable to static cameras only, whereas other model-based approaches account for both object and global camera motion [3, 4, 5, 6, 7]. The *object motion* corresponds to the observed motion of an object on the image plane when the camera is stationary. The *global motion* corresponds to the observed motion of an object on the image plane when, in the 3D world, the object is stationary and the camera moves. Global motion information is extracted from the coherent motion of the background when moving objects can be considered outlier motions, e.g. they move on a common plane that is related across frames by a homography [3, 4, 5, 7], which models rotation, zoom, any motion with respect to a planar surface and any combination of these. Model-based predictors for moving cameras use transformations among ground planes between consecutive frames that update over time [5] or 3D models to account for interacting objects using global and object motion, geometrical constraints and a reversible jump Markov chain Monte Carlo particle filter [6].

The *camera pose* may also be an important constraint for predictors as some methods are applicable only when the scene is seen by a top-down looking camera mounted on a drone and thus can be considered planar [3, 4, 7]. This assumption simplifies the explicit decoupling of camera and object motion by first estimating a frame-by-frame transformation (e.g. homography or registration) between planes and then subtracting the camera motion from the observed motion. Methods and dataset (e.g. a trajectory Forecasting Benchmark [19]) often account only for static cameras or from top-down view cameras on high-altitude stationary drones and are therefore limited methods to very specific scenarios.

In this paper, we propose a real-time object predictor that disentangles the motion of objects from that of an uncalibrated camera that moves at ground-level, a typical scenario for first-person vision and autonomous robots. The proposed model accurately predicts the location of objects without assuming that they move on a common plane. We remove the typical planarity constraint by approximating the observed camera motion on the image plane with a rotation across frames. The proposed predictor is resilient to local motions and low frame rate videos. Experiments show that the proposed prediction model outperforms alternative approaches by 60% in accuracy when predicting 30 frames with moving cameras. We also validate our predictor in a multi-object tracker and show an improvement of 3.6 accuracy points in moving-camera scenarios.

2. PREDICTION MODEL

Let $\mathbf{I}_k$ be a frame at time k and $\mathbf{x}_k = (u, v, 1)$ be the position, in homogeneous coordinates, of an object on the image plane, where u and v are the horizontal and vertical position of the center of the object, respectively. Our goal is to determine $\tilde{\mathbf{x}}_{k+T_F}$, the prediction of $\mathbf{x}_{k+T_F}$, where $T_F \geq 1$, given $T_P \geq 2$ past observations.

To facilitate the prediction of the object location, we decompose the motion observed on the image plane seen from a moving camera into global and object motion. The *global motion* can be inferred from the coherent motion of the background [4]. We propose GM, an object motion predictor that models the global background motion between two frames, $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$, with a homography, $\mathbf{H}_{k|k-1}$, assuming that between consecutive frames the global motion can be

Table 1. Object motion predictors. Key - Ref: reference; CS: coordinate system; A: approach; L: linear; NL: non linear; CM: robust to camera motion; FS: robust to frame skipping; NC: works without camera calibration; NS: not scene specific; FO: objects can move freely in the scene; Pro: proposed method.

Type	Ref	Strategy	CS	A	CM	FS	NC	NS	FO
data driven	[1]	learns typical motion patters using clustering	2D	NL			✓	✓	✓
	[2]	accounts for person-to-person interactions with LSTMs	2D	NL			✓	✓	✓
	[8]	accounts for person-to-person interactions and obstacles with LSTMs	2D	NL			✓	✓	✓
	[9]	handles long-term occlusions with LSTM	2D	NL			✓	✓	✓
model based	[10]	probabilistic multimodal approach	2D	NL			✓	✓	✓
	[11]	online learning for prediction of objects and groups of objects	2D	NL			✓	✓	✓
	[12]	pedestrian trajectory prediction	2D	NL			✓	✓	✓
	[13]	Brownian model	2D	L			✓	✓	✓
	[14]	Markov Chain Monte Carlo data association	2D	L			✓	✓	✓
	[15]	linear modelling with Gaussian noise	2D	L			✓	✓	✓
	[16]	linear modelling with Gaussian noise	2D	L			✓	✓	✓
	[17]	accounts for perspective and frame-rate	2D	L			✓	✓	✓
	[18]	accounts for perspective and frame-rate	2D	L			✓	✓	✓
	[3]	frame registration (aerial video) for camera motion estimation	2D	L	✓		✓		
	[4]	homography (aerial videos) camera motion estimation	2D	L	✓		✓		
	[5]	ground plane prediction (homography) for camera motion estimation	2D	L	✓				
	[6]	geometry priors (requires an RGB-D camera) for camera motion estimation	3D	L	✓			✓	✓
	[7]	frame-registration (aerial videos) for camera motion estimation	2D	L	✓		✓		
	Pro	decouples apparent object and camera motion (homography)	2D	L	✓	✓	✓	✓	✓

approximated with a camera rotation in first-person-view scenarios (i.e. camera translation effects are negligible). For the prediction, we approximate $\mathbf{H}_{k+1|k} \approx \mathbf{H}_{k|k-1}$ assuming that the global motion between k and $k+1$ is similar to that between $k-1$ and k. Therefore, given $\mathbf{x}_k$, the position of an object in $\mathbf{I}_k$, if the camera moves and the object is static in the real world, the predicted position of the same object in $\mathbf{I}_{k+1}$ is

$$\tilde{\mathbf{x}}_{k+1} = \frac{1}{\alpha_k}\mathbf{H}_{k|k-1}\,\mathbf{x}_k, \tag{1}$$

where $\alpha_k = \langle \mathbf{h}_3, \mathbf{x}_k^{\lambda} \rangle$ is a normalization factor; $\mathbf{h}_3$ is the third row of $\mathbf{H}_{k|k-1}$ and $\langle \cdot \rangle$ is the dot product. To estimate $\mathbf{H}_{k|k-1}$, we selectively detect and track across frames keypoints on the background (see Fig. 1).

Let the set of keypoints detected in $\mathbf{I}_{k-1}$ be

$$\mathbb{P}_{k-1} = \{\mathbf{p}_{k-1}^n = (u, v)\}, \tag{2}$$

where u and v are the horizontal and vertical coordinates of keypoint n on the image plane. We use as keypoint detector *good features to track* [20]. The presence of moving objects, which are outlier motions with respect to the global motion estimation, is an important aspect to model in first-person view. We apply a binary *filtering mask* to discard keypoints on moving objects. We generate the binary mask by combining the results of a classifier that identifies the bounding boxes of candidate moving objects of pre-defined classes (i.e. people and cars) and extend each bounding box by a margin c to account for inaccurate borders in the object detection results.

We then track the keypoints and localize them in $\mathbf{I}_k$ using a sparse iterative version of the Lucas-Kanade optical flow in pyramids [21]. Tracking generates two sets of matched keypoints: $\mathbb{P}'_{k-1}$ and $\mathbb{P}'_k$, with $|\mathbb{P}'_{k-1}| = |\mathbb{P}'_k|$ where $|\cdot|$ is the cardinality of a set. Note that some keypoints might be lost during tracking, $|\mathbb{P}'_{k-1}| \leq |\mathbb{P}_{k-1}|$. The filtered set of correspondent keypoint pairs, $\mathbb{P}''_{k-1}$ and $\mathbb{P}''_k$, are obtained after applying the mask filtering.

As the number of tracked keypoints decreases over time due to occlusions, tracking errors and corresponding 3D points exiting the field of view of the first-person-view camera, we keep a balanced spatial distribution of keypoints [22] by dividing the frame into $N_u \times N_v$ equally-sized cells and triggering a keypoint detection process in cells with fewer than N_m keypoints. When a keypoint detection process is triggered, new keypoints are detected, $\mathbb{P}_k^{(i,j)}$, only outside the filtering mask; where $i \in [1, N_u]$ and $j \in [1, N_v]$ are the horizontal and vertical indices of the cells. The new set of detected keypoints is $\mathbb{P}_k^* = \bigcup_{\forall i,j} \mathbb{P}_k^{(i,j)}$, where $\bigcup$ is the union operator.

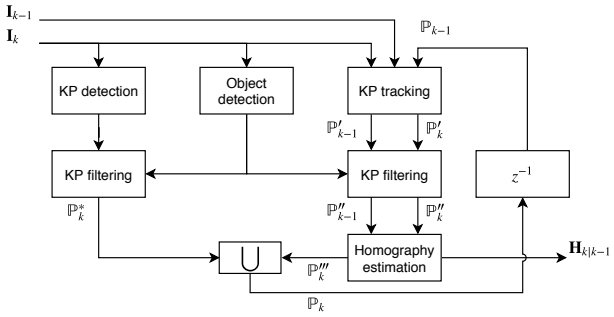

Fig. 1. Block diagram of the proposed homography estimation pipeline.

We then compute the homography, $\mathbf{H}_{k|k-1}$, with the filtered and matched keypoints, $\mathbb{P}''_{k-1}$ and $\mathbb{P}''_k$, by calculating the transformation that relates the position of the keypoints in these sets using random sample consensus (RANSAC) [23, 24]. RANSAC generates the set of inliers keypoints, $\mathbb{P}'''_k$, by discarding matched keypoints that follow a different (homography) transformation than the rest. Finally, the newly detected keypoints, $\mathbb{P}^*_k$, are added to the set of inlier keypoints as $\mathbb{P}_k = \mathbb{P}'''_k \cup \mathbb{P}^*_k$, which will be used for the next frame.

Then, the *object motion* over the past T_P observed locations can be estimated as

$$\dot{\mathbf{x}}_{k|k-T_P+1} = \frac{1}{T_P - 1}\sum_{i=0}^{T_P-2}(\mathbf{x}_{k-i} - \mathbf{H}_{k-i|k-i-1}\mathbf{x}_{k-i-1}), \tag{3}$$

and assuming that the motion of the object in the 3D world will be similar in the near future, we can iteratively predict the object location as

$$\tilde{\mathbf{x}}_{k+1} = \frac{1}{\alpha_k}\mathbf{H}_{k|k-1}\mathbf{x}_k + \dot{\mathbf{x}}_{k|k-T_P+1}. \tag{4}$$

In the next section we validate the proposed predictor.

3. VALIDATION

We compare the proposed method, GM, with six other predictors, we quantify the robustness to noisy observations and to frame rate reduction, and the benefit of GM for object tracking.

The six other predictors we compare against are: a predictor based on a Long Short-Term Memory (LSTM) [9]; two state-of-the-art predictors based on $\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_k + \dot{\mathbf{x}}_{k|k-T_P+1}$, namely a linear motion predictor (LP) [15] where

$$\dot{\mathbf{x}}_{k|k-T_P+1} = \frac{1}{T_P - 1}\sum_{i=0}^{T_P-2}(\mathbf{x}_{k-i} - \mathbf{x}_{k-i-1}), \tag{5}$$

and an exponentially weighted motion predictor (EM) [10] where

$$\dot{\mathbf{x}}_{k|k-T_P+1} = \frac{1}{\sum_{i=0}^{T_P-2}(\rho)^i}\sum_{i=0}^{T_P-2}(\rho)^i(\mathbf{x}_{k-i} - \mathbf{x}_{k-i-1}), \tag{6}$$

with $\rho = 0.95$; a linear regressor (LR) where $\tilde{\mathbf{x}}_{k+1} = \mathbf{m}\,\mathbf{x}_k + \mathbf{b}$, with $\mathbf{m}$ and $\mathbf{b}$ learned online for each object from its past locations; a homography-based (SH) method [4] and, as reference, a static-object prior-knowledge method (SP) with $\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_k$. In addition,

Table 2. Prediction error on moving-camera sequences (average and its standard deviation on the test dataset). Key – T_P: number of past observed locations; T_F: number of future locations to predict; SP: static object prior knowledge; LP: linear prediction; EM: exponentially-weighted prediction; LR: linear regressor; SH: simple homography-based predictor; GMG: proposed global motion with ground masking; GM: proposed global motion prediction; *: at least an order of magnitude larger (and therefore not reported). The lower the number the better the performance. Best and second best performing methods are shown in bold and italic, respectively.

T_P	T_F	SP		LP		EM		LR		LSTM		SH		GMG		GM	
2	1	7.3	(8.2)	**2.1**	**(4.2)**	**2.1**	**(4.2)**	**2.1**	**(4.2)**	6.1	(7.0)	2.8	*	*2.2*	*(4.2)*	*2.2*	*(4.2)*
	10	35.0	(47.4)	*13.7*	*(19.6)*	*13.7*	*(19.6)*	*13.7*	*(19.6)*	27.3	(36.9)	69.9	*	16.0	(26.4)	**13.0**	**(16.8)**
	20	60.3	(81.2)	*31.2*	*(40.5)*	*31.2*	*(40.5)*	*31.2*	*(40.5)*	47.4	(62.9)	194.6	*	31.0	(42.6)	**25.1**	**(31.9)**
	30	80.5	(106.1)	50.7	(62.2)	50.7	(62.2)	50.7	(62.2)	64.8	(82.7)	292.7	*	*46.1*	*(61.3)*	**37.5**	**(46.0)**
10	1	7.2	(8.2)	*2.8*	*(3.4)*	**2.7**	**(3.4)**	5.2	(5.8)	5.3	(6.5)	10.9	*	3.5	(5.2)	2.9	(3.7)
	10	35.0	(46.8)	15.4	(18.7)	14.9	(18.3)	17.6	(20.2)	25.0	(35.5)	55.4	*	*11.5*	*(18.8)*	**9.5**	**(14.2)**
	20	59.9	(79.4)	32.0	(38.0)	31.3	(37.5)	34.1	(39.2)	44.0	(60.5)	140.3	*	*19.2*	*(30.4)*	**16.0**	**(24.2)**
	30	79.4	(101.8)	49.7	(57.6)	49.0	(57.0)	51.9	(58.8)	59.9	(78.2)	199.8	*	*26.7*	*(39.8)*	**22.3**	**(33.0)**
20	1	7.2	(8.2)	3.6	(3.8)	*3.3*	*(3.7)*	12.1	(11.5)	5.6	(6.5)	11.1	*	3.6	(5.6)	**3.0**	**(3.9)**
	10	34.9	(46.2)	18.4	(21.7)	17.1	(20.4)	26.1	(26.8)	27.1	(35.9)	63.4	*	*11.1*	*(18.6)*	**9.4**	**(15.1)**
	20	59.3	(76.8)	35.7	(41.5)	33.8	(39.7)	42.8	(45.5)	49.6	(57.2)	145.4	*	*17.8*	*(28.8)*	**15.2**	**(24.8)**
	30	78.8	(99.3)	52.9	(60.7)	50.8	(58.7)	59.8	(64.2)	68.6	(76.3)	200.5	*	*24.5*	*(37.5)*	**20.6**	**(32.3)**
30	1	7.2	(8.2)	4.0	(4.3)	*3.5*	*(3.9)*	19.7	(18.5)	5.9	(6.4)	11.7	*	3.6	(5.3)	**3.1**	**(4.2)**
	10	34.6	(45.3)	20.3	(23.7)	18.1	(21.5)	34.2	(33.4)	28.2	(34.8)	53.6	*	*11.3*	*(18.6)*	**9.7**	**(16.2)**
	20	59.2	(76.0)	38.0	(44.3)	35.1	(41.1)	50.8	(51.7)	49.4	(59.1)	193.3	*	*18.0*	*(29.1)*	**15.5**	**(25.9)**
	30	78.8	(99.1)	55.4	(65.0)	52.0	(60.8)	67.2	(71.1)	65.1	(81.4)	247.5	*	*24.6*	*(37.9)*	**20.8**	**(33.3)**

we compare with GMG, a variation of the proposed method that assumes that the whole scene is a plane and all objects lie on that common ground plane, similarly to [4], by masking pixels that are in the estimated ground plane that is defined as the convex hull between the bottom corners of the detections and the bottom corners of the frame.

To compare the methods fairly and only on the accuracy of their prediction, we use the past T_P ground-truth locations of an object to predict its future T_F locations. If λ is the object index, the ground truth annotations, $\hat{\mathbb{X}} = \left\{\hat{\mathbf{x}}_k^\lambda | \forall\lambda, \forall k\right\}$, of the training dataset includes both static and moving objects. We quantify the prediction error as:

$$MSE = \frac{1}{\Lambda K'} \sum_{\lambda=1}^{\Lambda} \sum_{k'=k_s^\lambda+T_P}^{k_e^\lambda-T_F+1} \sum_{k=k'}^{k'+T_F-1} ||\hat{\mathbf{x}}_k^\lambda - \tilde{\mathbf{x}}_k^\lambda||_2^2, \quad (7)$$

where Λ is the total number of objects in the video, k_s^λ and k_e^λ are the first and last frame where object λ is visible by the camera, $K' = T_F \sum_{\lambda=1}^{\Lambda} (k_e^\lambda - k_s^\lambda - T_F - T_P + 2)$ is the total number of predictions within the video, and $||\cdot||_2$ is the L2-norm.

For the good-features-to-track detector [20] and for the sparse tracker [21] we use the default parameters of the OpenCV implementation (version 3.4.1): 50 as maximum number of corners, 0.01 as quality level and 10 pixels as minimum distance for the detector; and 21×21 as window size, three maximum levels of the pyramid and 0.001 as minimum eigenvalue threshold for the tracker. We calculate the homography from the set of correspondent keypoints with the OpenCV implementation and default parameters. The margin in the masking is $c = 0.05$. The minimum number of keypoints per cell is set to $N_m = 20$. For creating the filtering mask, we use SDP [25] to detect only humans.

We use three publicly available *datasets*: *Multiple Object Tracking Benchmark* 2015 (MOTB15 [26]), 2016 (MOTB16 [27]) and 2017 (MOTB17 [28]). These datasets are composed of moving sequences recorded from first-person-vision and static cameras. For the first three experiments, we build a *training*, *validation* and *test* dataset from the MOTB training sequences aiming to balance the number of annotations between static and moving cameras and do not use the same video in different subsets. The *training dataset* accounts for 44% of the dataset and it is composed of ADL-Rundle-8, ETH-Bahnhof, ETH-Sunnyday, KITTI-13, KITTI-17, PETS09-S2L1, TUD-Campus, TUD-Stadtmitte, MOT16-04, MOT16-10, MOT17-04 and MOT17-10. The *validation dataset* accounts for 16% of the dataset and it is composed of MOT16-05, MOT16-09, MOT17-05 and MOT17-09. The *test dataset* accounts for 40% of the dataset and it is composed of Venice-2, MOT16-02, MOT16-11, MOT16-13, MOT17-02, MOT17-11, MOT17-13.

Table 2 compares the prediction accuracy of GM against that of the other algorithms. LP, EM, LR and LSTM accumulate prediction errors when the prediction time is long ($T_F = \{10, 20, 30\}$) for any T_P, except for $T_P = 2$ where they obtain competitive results. These methods obtain an average prediction error of 55.4, 52.0, 67.2 and 65.1 pixels when predicting over 30 frames and having observed 30 past frames. GM obtains consistently the best result followed by GMG except for ($T_P = \{2, 10\}, T_F = 1$) where they are slightly outperformed by LP and EM. In general, the larger T_P (the more past frames are observed), the lower the prediction error, with a reduction of the improvement with $T_P = \{20, 30\}$. SH has large prediction errors due to the lack of a constraint for maintaining a spatial distribution of keypoints and of a masking procedure to eliminate outlier local motions.

Regarding to the robustness to detection errors, Figure 2 shows the object prediction accuracy in the presence of Gaussian noise of varying standard deviation in the past T_P observed object locations. While LR and LSTM are more robust to noise in relative terms, GMG and GM outperform the rest in absolute values when the standard deviation of the noise is lower than 20.

Regarding to the robustness to frame-rate reduction, Figure 3 shows the prediction errors when downsampling the frame rate. To constrain the temporal observation when the video frame rate decreases, we select T_P and T_F as $T_P = \lceil \frac{T'_P F}{\gamma} \rceil$ and $T_F = \lceil \frac{T'_F F}{\gamma} \rceil$ where T'_P and T'_F are the past/future number of seconds to observe/predict and F is the original frame rate of the video. When $\gamma = 1$ (50% frame rate reduction), GM has the lowest absolute prediction error with an error reduction of 41% with respect to the next best-performing methods (LP and EM), and 58% with respect

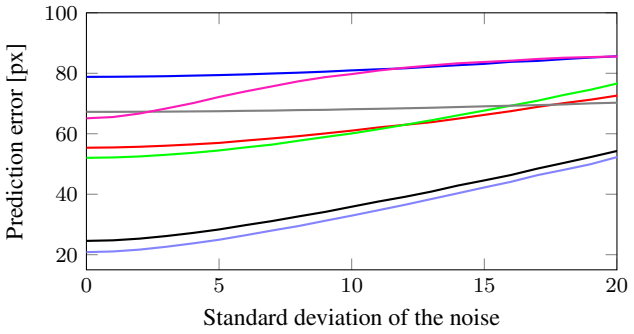

Fig. 2. Average prediction error in all moving-camera videos of the testing dataset for the next $T_F = 30$ frames when the observations over the past $T_P = 30$ frames are contaminated by Gaussian noise of varying standard deviation. For better visualization, not showing SH as its error is at least an order of magnitude larger. KEY – px: pixels; SP —, LP —, EM —, LR —, LSTM —, GMG — and GM —.

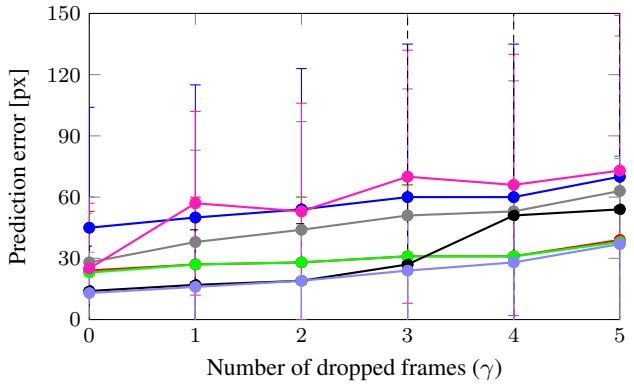

Fig. 3. Prediction errors (average and its standard deviation) in all moving-camera videos of the testing dataset for the next 0.5 s when observing their locations in the past 0.5 s. For better visualization, not showing SH as its error is at least an order of magnitude larger. KEY – px: pixels, SP —, LP —, EM —, LR —, LSTM —, GMG — and GM —.

to the subsequent next best-performing methods (LR and SH). When $\gamma = 4$ the error reduction is of 10% with respect to the next best-performing methods (LP and EM), and 47% with respect to the subsequent next best-performing methods (LR and SH). EM and LP perform similarly (3% and 5% larger prediction error) to GM when $\gamma = 5$ as GM accumulate global motion estimations errors over time. Using only 25% of the original frame rate ($\gamma = 3$), GM obtains comparable prediction accuracy to LP, EM, LR and LSTM at the original frame rate ($\gamma = 0$). These results indicate that the proposed method allows one to reduce the camera acquisition rate while still obtaining a lower prediction error compared to other algorithms.

We compare the processing speed of the methods under comparison and the proposed method with no code optimization in the testing dataset. All experiments are executed in a computer with an Intel i7 microprocessor with 16GB of RAM. The proposed method achieves average processing speed faster than 28 frames per second. Methods that do not perform image processing techniques (LSTM, LP, EM and SP) compute the predictions in less than 1 millisecond per frame. SH works at an average of 44 frames per second.

Table 3. Influence of different predictors in tracking performance (average MOTA and MOTP and their standard deviation on the MOTB17 training dataset). KEY – Ca: camera motion; M: moving-camera sequences only; C: complete dataset; LP: linear prediction [15]; SH: simple homography-based predictor [4]; GMG: proposed global motion with ground masking; GM: proposed global motion prediction. The higher the number the better the performance. Best and second best performing methods are shown in bold and italic, respectively.

Metric	Ca	LP		SH		GMG		GM	
MOTA	M	56.4	(0.5)	53.0	(0.4)	*58.1*	*(0.5)*	**60.0**	**(0.2)**
	C	62.6	(0.2)	61.4	(0.2)	*63.2*	*(0.2)*	**64.0**	**(0.1)**
MOTP	M	81.1	(0.0)	80.9	(0.1)	*81.5*	*(0.1)*	**81.7**	**(0.1)**
	C	*83.7*	*(0.1)*	83.6	(0.1)	**83.9**	**(0.1)**	**83.9**	**(0.1)**

Finally, we test the proposed predictor in a real application for first-person vision. We embed the predictor in a state-of-the-art multi-object tracker, the EA-PHD-PF [18], and compare the tracking performance against different prediction models. Table 3 shows the tracking performance with different prediction models on the MOTB17 training dataset measured as the average Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) [29] on five tracking runs with default parameters and SDP [25] as detector. When GM is used as prediction model, the highest MOTA and MOTP scores are achieved with moving cameras: GM allows the tracker to improve 3.6 MOTA points and 0.6 MOTP points compared to using LP as prediction model. This improvement in accuracy and precision are meaningful using the confidence intervals that depend on the accuracy of the ground-truth annotation [30]. An in-depth description of the integration of GM in the tracker and an extensive analysis of the results are available in [31].

4. CONCLUSION

We presented GM, an object motion predictor that is aware of the global camera motion. The proposed predictor does not require camera calibration parameters, the presence of planar surfaces, or prior knowledge about scene and objects. GM considerably reduces the prediction error compared to the state-of-the-art predictors when predicting over 30 frames. Moreover, GM outperforms the state-of-the-art models while processing fewer frames, thus allowing one to intentionally reduce the video frame rate and hence the energy consumption, an important aspect for first-person vision. Finally, we showed that when GM is integrated into a tracker, its performance increases by 3.6 MOTA points in moving-camera scenarios.

5. REFERENCES

[1] D. Vasquez and T. Fraichard, "Motion prediction for moving objects: a statistical approach," in *Proc. IEEE Int. Conf. Robotics Autom.*, New Orleans, LA, Apr. 2004, vol. 4, pp. 3931–3936 Vol.4.

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, June 2016, pp. 961–971.

[3] G. Máttyus, C. Benedek, and T. Szirányi, "Multi target tracking on aerial videos," in *Proc. ISPRS Workshop*, Istanbul, Turkey, 2010.

[4] S. Li and D. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *Proc. Association for the Advancement of Artificial Intelligence*, San Francisco, CA, June 2017, pp. 4140–4146.

[5] J. Arrospide, L. Salgado, M. Nieto, and R. Mohedano, "Homography-based ground plane detection using a single on-board camera," *Trans. IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 149–160, 2010.

[6] W. Choi, C. Pantofaru, and S. Savarese, "A general framework for tracking multiple people from a moving camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1577–1591, 2013.

[7] S. Lankton and A. Tannenbaum, "Improved tracking by decoupling camera and target motion," in *Proc. SPIE*, San Jose, CA, 2008, pp. 6811–6819.

[8] H. Sommer J. Nieto R. Siegwart C. Cadena M. Pfeiffer, G. Paolo, "A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments," in *Proc. IEEE Int. Conf. Robotics Autom.*, Brisbane, Australia, May 2018, pp. 1–8.

[9] M. Babaee, Z. Li, and G. Rigoll, "Occlusion handling in tracking multiple people using RNN," in *Proc. IEEE Conf. Image Process.*, Athens, Greece, Oct 2018, pp. 2715–2719.

[10] V. Akbarzadeh, C. Gagne, and M. Parizeau, "Target trajectory prediction in ptz camera networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, June 2013, pp. 816–822.

[11] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, June 2012, pp. 1918–1925.

[12] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, June 2011, pp. 1345–1352.

[13] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. IEEE Int. Conf. Robotics Autom.* IEEE, 2002, vol. 1, pp. 695–701.

[14] Q. Yu, G. Medioni, and I. Cohen, "Multiple Target Tracking Using Spatio-Temporal Markov Chain Monte Carlo Data Association," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HA, USA, 2007, pp. 1–8.

[15] K. Shafique, Mun Wai Lee, and N. Haering, "A rank constrained continuous formulation of multi-frame multi-target tracking problem," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.

[16] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, Sept. 2011.

[17] E. Maggio, M. Taj, and A. Cavallaro, "Efficient multitarget visual tracking using random finite sets," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1016–1027, Aug. 2008.

[18] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Online multi-target tracking with strong and weak detections," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 84–99.

[19] A. Sadeghian, V. Kosaraju, A. Gupta, S. Savarese, and A. Alahi, "Trajnet: Towards a benchmark for human trajectory prediction," *arXiv preprint*, 2018.

[20] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Ithaca, NY, 1994, pp. 593–600.

[21] J. Bouguet, "Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the algorithm," *Internal Report Intel Corporation*, vol. 5, no. 1-10, pp. 4, 2001.

[22] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, no. 3, pp. 219–238, 2016.

[23] M.A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[24] R. Szeliski, "Image alignment and stitching: A tutorial," *Trans. on Foundations and Trends in Comp. Graph. and Vis.*, vol. 2, no. 1, pp. 1–104, 2006.

[25] F. Yang, W. Choi, and Y. Lin, "Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, June 2016, pp. 2129–2137.

[26] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, 2015.

[27] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831 [cs]*, 2016.

[28] "MOT17: A benchmark for multi-object tracking," `https://motchallenge.net/data/MOT17/`, 2016, [Online; accessed 31^{st}-January-2019].

[29] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT Metrics," *Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 246–309, May 2008.

[30] R. Sanchez-Matilla and A. Cavallaro, "Confidence intervals for tracking performance scores," in *Proc. IEEE Conf. Image Process.*, Athens, Greece, Oct. 2018, pp. 246–250.

[31] R. Sanchez-Matilla and A. Cavallaro, "Accurate prediction for causal multi-target tracking," submitted to *IEEE Conf. Image Process.*, 2019.