Tracking of 2D or 3D Irregular Movement by a Family of Unscented Kalman Filters

Junli Tao and Reinhard Klette

Abstract—The paper reports about the design of an object tracker which utilizes a family of unscented Kalman filters, one for each tracked object. This is a more efficient design than having one unscented Kalman filter for the family of all moving objects. The performance of the designed and implemented filter is shown by using simulated movements, and also for object movements in 2D an 3D space.

Index Terms—unscented Kalman filter, detection-by-tracking, 2D/3D tracking.

1 Introduction

Object tracking is an important task for many applications, such as for robot navigation, surveillance, automotive safety, or video content indexing. Based on trajectories obtained through tracking, some advanced behaviour analysis can be applied. For instance, the pedestrian trajectory can be analysed to warn a driver if trajectories of the vehicle and of the pedestrian are potentially intersecting.

For multiple object tracking, tracking-by-detection methods are the most popular algorithms. A detector is used in each image frame to obtain candidate objects. Then, with a data-association procedure, all the candidates are matched to the existing trajectories as known up to the previous frame. Any unmatched candidate starts a new trajectory. Since there is no perfect detector that detects all objects without any false positives and false negatives, sometimes objects are missed (i.e. they appear in the image but are not detected), or background windows are incorrectly detected as being objects. Such false-positive or false-negative detections increase the difficulty of tracking.

Occlusion by other objects or background is one main reason that causes the detection to fail and increases the difficulty for tracking(e.g., identity switch). Some algorithms [22] [1] propose to track objects in the 2D image plane. The occlusion problem is handled either using part detectors and tracking detected body parts, or adopting instance-specific classifiers to improve performance of data assignment. However, tracking in 2D image plane itself increases the ambiguity for data association. A tall person nearby, and a small person far away, for example, may appear very close to each other in the image, and probably the small person is occluded by the tall one in some frames. but they are actually several meters away from each other. Thus, often, and also in this paper, stereo information is adopted to improve the tracking performance [6] [12] [14], and multiple pedestrians are tracked in 3D coordinates.

Tracking objects with irregular movements in 3D space is a challenging task due to totally unknown speed and direction. In this paper, the application of an *unscented Kalman filter* (UKF) is demonstrated which can also handle nonlinear (actually: fully irregular) trajectories in 3D space. For the original paper on UKF, see [19]. Similar work is proposed in [14]. But instead of modelling the motion of the vehicle and the pedestrians separately, we straight forward model the relative motion between them, and no ground plane is assumed, objects moving in 6 degree-of-freedom can be tracked properly. Different types of model are tested

• *Junli Tao and Reinhard Klette are with the Department of Computer Science, University of Auckland, Auckland, NZ. E-mail: jtao076@aucklanduni.ac.nz*

Manuscript received July 18, 2012; revised.

and compared in both simulation and real sequences.

2 RELATED WORK

Multiple object tracking attracts much of attention recently in computer vision research. Today, an update of the review [24] from 2006 should also include work such as in [1], [5], [6], [12], [13], [15]–[18].

Kalman filters (KF) are extensively adopted to deal with tracking task. KF is a recursive Bayesian filter, firstly, using motion information to predict possible position, followed by fusing the observation (detection) and predicted position. A linear Kalman filter is used for tracking (see, e.g., [24]) when movement can be approximated by linear models. Obviously, a linear model is not true for most of the cases. The *extended Kalman filter* (EKF) was designed [21] for handling a nonlinear model by linearizing functions using the Taylor expansion extensively. For example, an EKF has been used for *Simultaneous Localization and Mapping* (SLAM) [9], and for pedestrian tracking [23]. Particle filter were used to handle the task in [2]. Similar performance as EKF is reported in [6].

The UKF can handle a nonlinear model by using the *unscented transform* to estimate the first and second order moments of *sigma points*, which represent the distribution of a predicted state and predicted observations, and it appears that the UKF does this better than the EKF [8]. Thus, in this paper, an UKF is used for tracking multiple, irregularly moving objects in 3D space, which is a 'highly nonlinear' problem.

3 UNSCENTED KALMAN FILTER

The *unscented transform* (UT) is the core part that makes UKF able to handle nonlinear models. Let L be the dimensionality of the system state $\mathbf{x}_{t-1|t-1}$ at time $t-1$. If the system noise (process noise $\mathbf{Q}$ and measurement noises $\mathbf{R}$) is not additive, the state is augmented before UT. In our case, the random acceleration is introduce as the process noise, thus, the state augmented with a process noise vector, is denoted by $\mathbf{x}^{(\mathbf{a})}_{t-1|t-1}$, called *vectors*. The dimension of the augmented vector, depends on the process model, which is illustrated in Section 4.2. Let $\mathbf{x}_{t|t-1}$ denote the predicted state at time t when passing $\mathbf{x}_{t-1|t-1}$ through process function $\mathbf{f}$. Let $\mathbf{y}_{t|t-1}$ be the predicted observation at time t when passing $\mathbf{x}_{t|t-1}$ through observation function $\mathbf{h}$.

The UT works by sampling $2L+1$ *sigma vectors* $X^{(\mathbf{a})}_i$ in the augmented state space (following [19]), forming a matrix $\mathcal{X}^{(\mathbf{a})}$. The covariance matrix in augmented state space is denoted by $\mathbf{P}^{(\mathbf{a})}$. Let $\mathbf{P}^{(\mathbf{xx})}_{t-1|t-1}$ be the state covariance matrix (i.e. describing dependencies between components of a state $\mathbf{x}$). Formally,

$$
\begin{aligned}
X^{(\mathbf{a})}_0 &= \mathbf{x}^{(\mathbf{a})}_{t-1|t-1} \\
X^{(\mathbf{a})}_i &= \mathbf{x}^{(\mathbf{a})}_{t-1|t-1} + (\sqrt{(L+\lambda)\mathbf{P}^{(\mathbf{a})}_{t-1|t-1}})_i \\
&\quad \text{for } i = 1, 2, \ldots, L \\
X^{(\mathbf{a})}_i &= \mathbf{x}^{(\mathbf{a})}_{t-1|t-1} - (\sqrt{(L+\lambda)\mathbf{P}^{(\mathbf{a})}_{t-1|t-1}})_{i-L} \\
&\quad \text{for } i = L+1, L+2, \ldots, 2L \\
\mathcal{X}^{(\mathbf{a})} &= \begin{bmatrix} \mathcal{X}^{(\mathbf{s})} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathcal{X}^{(\mathbf{n})} \end{bmatrix}
\end{aligned}
$$

where λ is a positive real, a scaling parameter. These sigma vectors can be passed through a nonlinear function (e.g., $\mathbf{f}$, $\mathbf{h}$) one by one, thus defining transformed (i.e. new) sigma vectors (e.g., $\mathcal{X}^{(\mathbf{s})}_{t|t-1}$ and $\mathcal{Y}_{t|t-1}$ are obtained). The means $\mathbf{x}_{t|t-1}$ or $\mathbf{y}_{t|t-1}$ and covariance matrices $\mathbf{P}^{(\mathbf{xx})}_{t|t-1}$, or $\mathbf{P}^{(\mathbf{yy})}_{t|t-1}$, are obtained as follows, take $\mathbf{h}$ for example:

$$
\begin{aligned}
\mathcal{Y}_{t|t-1} &= \mathbf{h}(\mathcal{X}^{(\mathbf{s})}_{t-1|t-1}) \\
\mathbf{y}_{t|t-1} &\approx \sum_{i=0}^{L} W^{(m)}_i Y_i \\
\mathbf{P}^{(\mathbf{yy})}_{t|t-1} &\approx \sum_{i=0}^{L} W^{(e)}_i (Y_i - \mathbf{y}_{t|t-1})(Y_i - \mathbf{y}_{t|t-1})^T
\end{aligned}
$$

with constant weights $W^{(.)}_i$. Details are given in [19].

The UKF is illustrated as follows. At first we initialize the state $\mathbf{x} = \mathbf{x}_0$ and state covariances $\mathbf{P}^{(\mathbf{xx})} = \mathbf{P}^{(\mathbf{xx})}_0$. For the augmented vectors, let

$$
\begin{aligned}
\mathbf{x}^{(\mathbf{a})} &= (\mathbf{x}^T \ \mathbf{0}^T)^T \\
\mathbf{P}^{(\mathbf{a})} &= diag(\mathbf{P}^{(\mathbf{xx})}, \mathbf{Q})
\end{aligned}
$$

where $\mathbf{Q}$ denotes the process-noise covariance matrix. Details about $\mathbf{Q}$ are given in Section 4.2.

For $t \in (1, \ldots, \infty)$, we calculate sigma vectors as follows:

$$\begin{aligned}\mathcal{X}^{(\mathbf{a})}_{t-1|t-1} &= (\mathbf{x}^{(\mathbf{a})}_{t-1|t-1}, \\ &\quad \mathbf{x}^{(\mathbf{a})}_{t-1|t-1} + \gamma\sqrt{\mathbf{P}^{(\mathbf{a})}_{t-1|t-1}}, \\ &\quad \mathbf{x}^{(\mathbf{a})}_{t-1|t-1} - \gamma\sqrt{\mathbf{P}^{(\mathbf{a})}_{t-1|t-1}})\end{aligned}$$

where $\gamma = \sqrt{\lambda + L}$. The process update is defined as follows:

$$\begin{aligned}\mathcal{X}^{(\mathbf{s})}_{t|t-1} &= f(\mathcal{X}^{(\mathbf{s})}_{t-1|t-1}, \mathcal{X}^{(\mathbf{n})}_{t-1|t-1}) \\ \mathbf{x}_{t|t-1} &= \sum_{i=0}^{2L} W_i^{(m)} X_i^{(\mathbf{s})} \\ \mathbf{P}^{(\mathbf{xx})}_{t|t-1} &= \sum_{i=0}^{2L} W_i^{(c)} (X_i^{(\mathbf{s})} - \mathbf{x}_{t|t-1})(X_i^{(\mathbf{s})} - \mathbf{x}_{t|t-1})^T\end{aligned}$$

We update the sigma vectors using

$$\begin{aligned}\mathcal{X}^{(\mathbf{s})}_{t|t-1} &= (\mathbf{x}_{t|t-1}, \\ &\quad \mathbf{x}_{t|t-1} + \gamma\sqrt{\mathbf{P}^{(\mathbf{xx})}_{t|t-1}}, \\ &\quad \mathbf{x}_{t|t-1} - \gamma\sqrt{\mathbf{P}^{(\mathbf{xx})}_{t|t-1}}) \\ \mathcal{Y}_{t|t-1} &= h(\mathcal{X}^{(\mathbf{s})}_{t|t-1}) \\ \mathbf{y}_{t|t-1} &= \sum_{i=0}^{2L} W_i^{(m)} Y_i\end{aligned}$$

and update the measurement covariance matrix as follows:

$$\mathbf{P}^{(\mathbf{yy})}_{t|t-1} = \sum_{i=0}^{2L} W_i^{(c)} (Y_i - \mathbf{y}_{t|t-1})(Y_i - \mathbf{y}_{t|t-1})^T + \mathbf{R}$$

where $\mathbf{R}$ is the assumed measurement noise covariance, depending on the observation model selected. Details are given in Section 4.2.

Altogether, the UKF is defined by

$$\begin{aligned}\mathbf{P}^{(\mathbf{xy})}_{t|t-1} &= \sum_{i=0}^{2L} W_i^{(c)} (X_i - \mathbf{x}_{t|t-1})(Y_i - \mathbf{y}_{t|t-1})^T \\ \mathcal{K}_t &= \mathbf{P}^{(\mathbf{xy})}_{t|t-1} (\mathbf{P}^{(\mathbf{yy})}_{t|t-1})^{-1} \\ \mathbf{x}_{t|t} &= \mathbf{x}_{t|t-1} + \mathcal{K}_t(\mathbf{y_t} - \mathbf{y_{t|t-1}}) \\ \mathbf{P}^{(\mathbf{xx})}_{t|t} &= \mathbf{P}^{(\mathbf{xx})}_{t|t-1} - \mathcal{K}_t \mathbf{P}^{(\mathbf{xy})}_{t|t-1} \mathcal{K}_t^T\end{aligned}$$

4 MULTIPLE OBJECT TRACKING

Following tracking-by-detection methods, which are popular for solving multiple-object tracking tasks, a detector is applied in each frame to generate *object candidates* which are outputs of the detector. One UKF is adopted for tracking one object separately, thus a group of detected pedestrians defines a family of UKFs to be processed simultaneously. Each UKF tracks one detected object. The predicted state of an UKF is used for data association; when an observation (of the tracked object) is available in the current frame then we update the predicted state by using the corresponding UKF.

4.1 Detection

Detection-by-tracking methods rely on evaluating rectangular *regions of interest*, we call them *object boxes* if positively identified as containing an object of interest. For pedestrian tracking, we adopt the popular *histogram of oriented gradients* (HOG) feature method and a *support vector machine* (SVM) classifier, originally introduced in [4]. HOG features describe the human profile by an oriented gradient histogram. An SVM classifier is able to handle high-dimensional and non-linear features (such as HOG features). It projects sample features into a high-dimensional space, and then finds a hyperplane to separate two classes. Instead of using a sliding window, regions of interest (i.e. inputs to the classifier) are selected by analysing calculated stereo information (depth and disparity maps), as proposed in [7].

Figure 1 shows several detection results in pedestrian sequence, dots(cyan) denotes the boxes centre that recognized as pedestrian, and the red rectangle denotes the final detection results. As can be seen in the results, the object boxes may contain background, shift from the object, or miss the pedestrians.

For the detection of *Drosophila larvae* (an example of 2D movement), thresholds and connected components are adopted to obtain one object box for each larvae. Several larvae detection results are shown in Fig. 2. As the scene is certain, the detection results are more reliable

when compare to pedestrian sequence. But no depth information is available here.

4.2 UKF-Based Object Tracking

As there is an unknown number of objects in a scene, the state-dimensionality would expand significantly if we would have decided to track all pedestrians in one UKF; in this case, the speed of tracking reduces dramatically when the scene is crowded with many detected objects around. Thus, we decided for one UKF for each detected object for tracking.

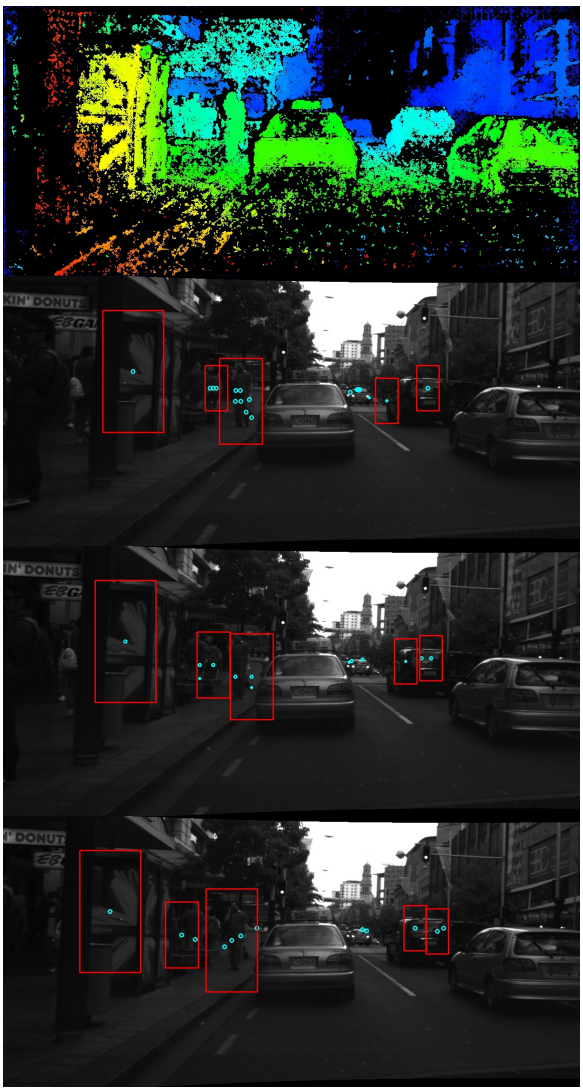

Fig. 1. The depth map on top uses a colour code for calculated distances; depth values are only shown at pixels where the mode filter accepts the given value. The lower image shows detected (coloured) object boxes.

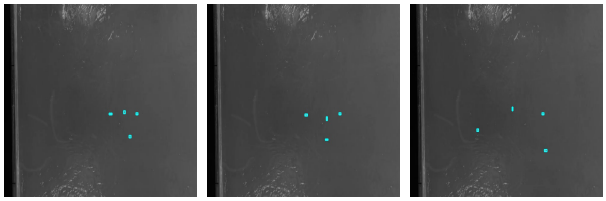

Fig. 2. Larvae detection results shown by (cyan) object boxes.

Choosing a proper model is important. In this subsection we offer three models for possible selection, namely 3D position (world coordinates) with velocity observed, denoted by 3DVT, 3D position without velocity observed, denoted by 3DT, and 2D position (image coordinates) without velocity observed, denoted by 2DT. These models are compared in Section 5.

4.2.1 *Both 3D models*

In the 3DVT model, the object is tracked in 3D world coordinates. Its 3D position (x, y, z) is the first part of the state. We also include the velocity (v_x, v_y, v_z). Thus, a state $\mathbf{x} = (x, y, z, v_x, v_y, v_z)^T$ is 6-dimensional.

Process Model. We assume constant velocity between adjacent frames, with Gaussian distributed noise acceleration $\mathbf{n_a} \in N(0, \Sigma_{n_a})$. The diagonal elements in Σ_{n_a} are set to be equal and denoted by $\sigma^2_{n_a}$. Thus,

$$\begin{aligned} \dot{x} &= x + (v_x + n_{ax}\Delta t)\Delta t, \quad \dot{v_x} = v_x + n_{ax}\Delta t \\ \dot{y} &= y + (v_y + n_{ay}\Delta t)\Delta t, \quad \dot{v_y} = v_y + n_{ay}\Delta t \\ \dot{z} &= z + (v_z + n_{az}\Delta t)\Delta t, \quad \dot{v_z} = v_z + n_{az}\Delta t \end{aligned}$$

where Δt is the time interval between subsequent frames.

Observation Model. An observation consists of the position (i_o, j_o) (say, the centroid of the detected object box in the left camera), disparity d of the detected object, and velocity (v_{ox}, v_{oy}, v_{oz}) in 3D coordinates. The usual pinhole-camera projection model is used to map 3D points into the image plane,

$$\begin{aligned} i_o &= fx/z, \quad j_o = fy/z \\ d &= fb/z, \quad v_{ox} = v_x \\ v_{oy} &= v_y, \quad v_{oz} = v_z \end{aligned}$$

where f denotes focal length, and b denotes the length of the baseline between two

rectified stereo cameras. In this case, $\mathbf{R} = diag(\sigma^2_{n_{mp}}, \sigma^2_{n_{mp}}, \sigma^2_{n_{mp}}, \sigma^2_{n_{mv}}, \sigma^2_{n_{mv}}, \sigma^2_{n_{mv}})$.

For the disparity d we select the mode in the disparity map in a fixed (e.g. 20×20) neighbourhood around the centroid (i_o, j_o) of the detected object box. 3D scene flow (v_{ox}, v_{oy}, v_{oz}) can be obtained by combining optic flow and stereo information [20].

As it is difficult to obtain high-quality scene flow as required for 3DVT, 3DT simplifies 3DVT model by excluding the scene flow in observation, and has the same process model as 3DVT. In this case, $\mathbf{R} = diag(\sigma^2_{n_{mp}}, \sigma^2_{n_{mp}}, \sigma^2_{n_{mp}})$.

4.2.2 The 2D model

If only monocular recording is available, the object is tracked in the 2D image plane only. The state $\mathbf{x} = (i, j, v_i, v_j)^T$ consists of position (i, j) and velocity (v_i, v_j).

Process Model. The same as for the 3D models. We assume constant velocity between subsequent frames with a Gaussian noise distribution for acceleration $\mathbf{n_a}$:

$$\dot{i} = i + \times(v_i + n_{a_i}\Delta t)\Delta t, \quad \dot{v_i} = v_i + n_{a_i}\Delta t$$
$$\dot{j} = j + \times(v_j + n_{a_j}\Delta t)\Delta t, \quad \dot{v_j} = v_j + n_{a_j}\Delta t$$

Observation Model. An observation consists of the central position (i_o, j_o) of an object box only, $i_o = i$ and $j_o = j$, resulting in $\mathbf{R} = diag(\sigma^2_{n_{mp}}, \sigma^2_{n_{mp}})$ for this case.

4.3 Data Association

As each object is tracked independently, data association by matching candidates to existing trajectories becomes important. If no match then we decide to initialize a new tracker.

Since object movements are continuous, the estimated velocity in the UKF can be used as a cue to localize the search area for finding the match object. For each trajectory, the possible location (i.e. (x_p, y_p, z_p) for 3D, and (i_p, j_p) for 2D) of the object in the current frame is predicted by process model used in the EKF. This location is used as a reference for searching potentially matching candidates in the current frame. Currently we simply match candidates based on shortest Euclidean distance and a given threshold τ.

One candidate might be matched with several trajectories if the Euclidean distance is below τ. Trajectories compete for the candidates, the closest wins finally. If a candidate is not matched to any trajectory, a new tracker is initialized. If a trajectory does not win any of the candidates, the tracker is propagated with the given prediction, and the new state is the predicted state, without being updated by an observation (because not available).

No object appearance description is used here for assigning an object to a trajectory. In general, the inclusion of appearance representation (e.g., a colour histogram, or an instance-specific shape model) improves the performance. However, this is out of the scope of this paper where we discuss the combination of different data-association methods.

5 Experiments

In this section, first, our three models (3DVT, 3DT, and 2DT) are tested in a simulated environment with different parameter sets. Second, our multiple-object tracking method is tested on real video sequences where (3D example) pedestrians are walking in inner-city scenes, or (2D example) larvae are moving on a flat culture dish.

5.1 Simulated Tracking

The three models defined in Section 4.2 are tested in a simulation environment in OpenGL. A cub is moving on a circular path around a 3D point with constant speed, as show in Figs. 3 and 4. Acceleration noise $\mathbf{n_a}$ with different covariance (e.g. $\sigma^2_{n_a} = 0.0001, 0.01, 1$), and measurement noise $\mathbf{n_m}$ with different covariance (e.g. $\sigma^2_{n_{mp}} = 10, 50, 100, \sigma^2_{n_{mv}} = 50, 100, 150$), are used to test and compare the three models' performance. The simulation environment is different for 2D and 3D models, where for 2D, positions are integral pixel coordinates in the image plane, but for 3D, position coordinates are reals. The radius of the circle in the 3D models is 10, while in the 2D model it is 50. In both environments, measurements are degraded by noise before sent to the UKF.

Figure 3 demonstrates the effect of $\sigma^2_{n_a}$, having fixed $\sigma^2_{n_{mp}}$ and $\sigma^2_{n_{mv}}$. Experiments show

that lager $\sigma^2_{n_a}$ values result in more unstable trajectories. Large $\sigma^2_{n_a}$ means that the process model produces a predicted state that is fluctuating with large magnitudes. Results show that $\sigma^2_{n_a} = 0.0001$ is a reasonable choice for 3D models. For the 2D model, a smaller $\sigma^2_{n_a}$ yields smooth estimation, but shifts are significant. A larger $\sigma^2_{n_a}$ value produces estimations that are more close to the true positions, but fluctuations are significant, for an experiment with $\sigma^2_{n_a} = 1$ for the 2D case. In general, 3DT and 3dDVT converge better than 2DT, while 3DT and 3dDVT show a similar performance. 3D models use stereo information rather than just a single image as for the 2D model, which also proves that stereo information can help to improve the tracking performance. As the measured 3D position is noisy, the measure of velocity is even more noisy; this appears to be the main reason for the observation that the inclusion of velocity cannot improve the performance.

Figure 4 shows results for the three models for different covariance values $\sigma^2_{n_{mp}}$ and $\sigma^2_{n_{mv}}$ of measurement noise. Significantly increasing measurement noise (i.e. higher uncertainty of observations) reduces the performance only slightly. This demonstrates that the UKF is a robust tracker to some degree, which is not vulnerable to detection uncertainties. As before, 3DT and 3dDVT converge better than 2DT, while 3DT and 3dDVT show a similar performance.

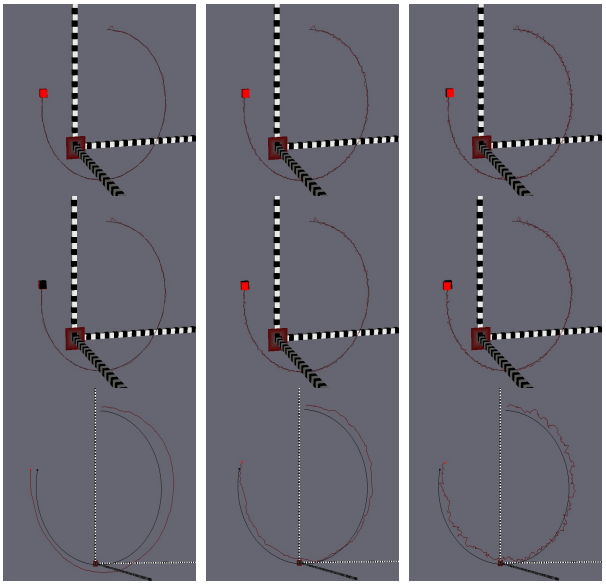

Fig. 3. Simulation results for variations in the variance of acceleration noise. From left to right, $\sigma^2_{n_a} = 0.0001$, 0.01, or 1, with fixed values $\sigma^2_{n_m} = \sigma^2_{n_{mp}} = 50$, and $\sigma^2_{n_{mv}} = 100$. From top to down, the tracking model is 3DVT, 3DT and 2DT respectively.

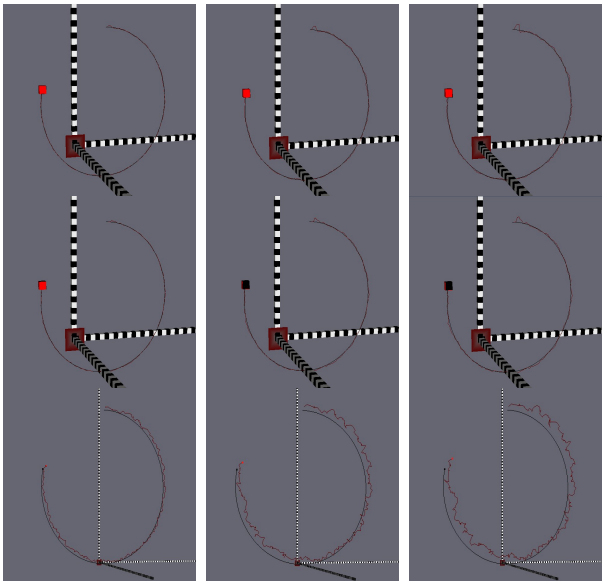

Fig. 4. Simulation results for variable variance of measurement noise. From left to right, $\sigma^2_{n_{mp}} = 10$, 50, or 100, $\sigma^2_{n_{mv}} = 50$, 100, 150, respectively, with fixed $\sigma^2_{n_a} = 0.0001$ for 3D models, $\sigma^2_{n_a} = 1$ for 2D models,. From top to down, the tracking model is 3DVT, 3DT and 2DT respectively.

5.2 Multiple Object Tracking in Real Data

In this section we report about the performance of UKF-supported tracking for multiple larvae using the 2DT model, and for multiple pedestrians in traffic scenes using the 3DT model. larvae sequence and pedestrian sequences are 30 and 15 frames per second.

Results for larvae tracking are shown in Fig. 5. As the velocity in the model is initialize by $(0, 0)$, the UKF-estimation is "slower" than the real speed of the larvae in the first 30 frames. The speed of convergence can be improved by increasing σ_{n_a}, but note that the larger the σ_{n_a} value is, the larger is the magnitude of fluctuation. The estimated trajectories follow "well" the larvae moving, mainly

because all larvae are properly detected in all frames. However, such a complete detection cannot be expected for pedestrian sequences. Next, we test the UKF for such "noisy detection results" for pedestrian sequences.

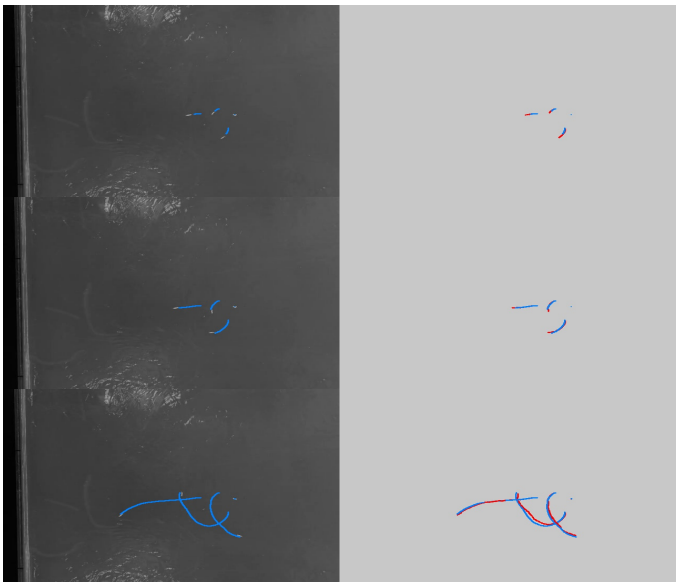

Fig. 5. 2D Tracking results of larvae sequences. From top to bottom: tracking results in Frames 26, 46, and 166 of one sequence. Red lines show the detected track, and white lines show the UKF-predicted track. Blue lines represent estimated trajectories. The left column is the original intensity image overlaid with estimated trajectories.

Results for pedestrian tracking are shown in Fig. 6. Objects are missing or shifting from time to time due to the clustered background (car in the traffic scene detected as a pedestrian), illumination variations (some pedestrians are not detected), or internal variations between objects (unstable detections). Our experiments verified that UKF-predictions can follow irregular moving pedestrians when detection fails for a few frames, and can even correct unstable detections.

The second frame in Fig. 6 shows that the undetected pedestrian is predicted correctly in the white object box, and successfully matched to a detected position in the third frame. The last frame in Fig. **??** demonstrates that displaced detections are corrected by the UKF. Using only the defined distance rule for data assignment, this appears to be insufficient, especially for the given detection results. A small threshold may lead to a mismatch (i.e. the detection fails to satisfy the rule), and a large threshold may lead to an identity switch (i.e. a pedestrian is matched to another pedestrian).

6 Conclusions

Assigning one unscented Kalman filter to each detected (moving) object simplifies the design and implementation of UKF-prediction of 2D or 3D motion. Experiments demonstrate the robustness of the chosen approach. This tracker only generates short-term tracks when detection is not reliable; long-term tracking should be possible by also introducing dynamic programming. For evaluating the performance on real-world (either 2D or 3D) applications, more extensive tests need to be undertaken, especially for the design and evaluation of quantitative performance measures. For example, measures discussed in [10] for evaluating visual odometry techniques might also be of relevance for the tracking case.

Acknowledgment

The authors thank *Simon Hermann* for providing his implementation of semi-global matching for stereo analysis, *Gabriel Hartmann* (both Auckland) for support regarding the unscented Kalman filter, and *Benjamin Risse* (Münster) for video data with moving Drosophila larvae.

References

[1] 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 Analysis Machine Intelligence*, **33**:1820–1833, 2011.

[2] Y. Cai, N. de Freitas, and J. J. Little. Robust visual tracking for multiple targets. In Proc *ECCV*, pages 107–118, 2006.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Analysis Machine Intelligence*, **25**:564–577, 2003.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Proc. *CVPR*, volume 1, pages 886–893, 2005.

[5] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust multiperson tracking from a mobile platform. *IEEE Trans. Pattern Analysis Machine Intelligence*, **31**:1831–1846, 2009.

[6] A. Ess, K. Schindler, B. Leibe, and L. Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *Int. J. Robotic Research*, **29**:1707–1725, 2010.

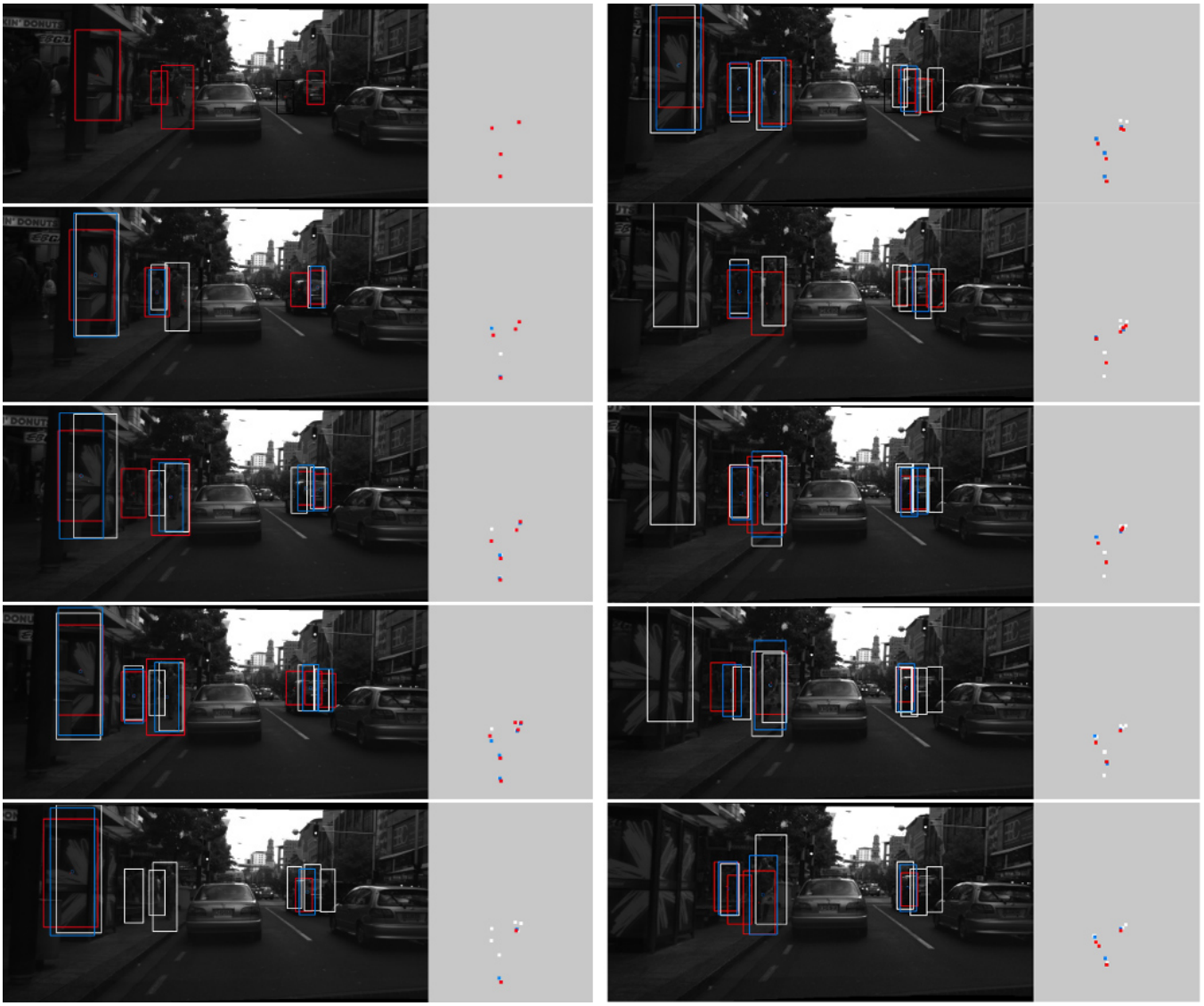

Fig. 6. Left: object boxes (red), UKF-predictions (white), and UKF-estimations (blue) are overlaid on the original intensity image. The black box is an invalid detection excluded by a disparity check. Right: coloured blocks denote the object boxes (red), UKF-predictions (white), and UKF-estimations (blue) in the XZ-plane in real-world coordinates.

[7] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *Int. J. Computer Vision*, **73**:41–59, 2007.

[8] G. Hartmann. Unscented Kalman filter sensor fusion for monocular camera localization. MSc thesis, Computer Science, Univ. Auckland, 2012.

[9] S. Huang. Convergence analysis for extended Kalman filter based SLAM. In Proc. *IEEE Int. Conf. Robotics Automation*, pages 412–417, 2006.

[10] R. Jiang, R. Klette, and S. Wang. Statistical modeling of long-range drift in visual odometry. In Proc. *CVVT, ACCV workshop*, LNCS 6469, pages 214–224, 2011.

[11] E. Kraft. A quaternion-based unscented Kalman filter for orientation tracking. In Proc. *Int. Conf. Information Fusion*, volume 1, pages 47–54, 2003.

[12] B. Leibe, K. Schindler, N. Cornelis, S. Member, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE Trans. Pattern Analysis Machine Intelligence*, **30**:1683–1698, 2008.

[13] W. F. Leven and A. D. Lanterman. Unscented Kalman filters for multiple target tracking with symmetric measurement equations. *IEEE Trans. Automatic Control*, **54**:370–375, 2009.

[14] M. Meuter, U. Iurgel, S. B. Park, and A. Kummert. The unscented Kalman filter for pedestrian tracking from a moving host. *IEEE Symp. Intelligent Vehicles*, pages 37–42, 2008.

[15] D. Mitzel, E. Horbert, A. Ess, and B. Leibe. Multi-person tracking with sparse detection and continuous segmentation. In Proc *ECCV*, LNCS 6311, pages 397–410, 2010.

[16] D. Mitzel, P. Sudowe, and B. Leibe. Real-time multi-person tracking with time-constrained detection. In Proc *BMVC*, pages 104.1–104.11, 2011.

[17] D. Mitzel and B. Leibe. Real-time multi-person tracking

with detector assisted structure propagation. In Proc *ICCV Workshops*, pages 974–981, 2011.
[18] M. M. Shaikh, W. Bahn, C. Lee, T. Kim, T. Lee, K. Kim, and D. Cho. Mobile robot vision tracking system using unscented Kalman filter. In Proc. *Int. Symp. System Integration*, pages 1214–1219, 2011.
[19] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In Proc. *IEEE Symp. AS-SPCC*, pages 153–158, 2000.
[20] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3D motion understanding. *Int. J. Computer Vision*, **95**:29–51, 2011.
[21] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report, Univ. North Carolina at Chapel Hill, 1995.
[22] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *Int. J. Computer Vision*, **75**:247–266, 2007.
[23] H. S. Yazdi and S. E. Hosseini. Pedestrian tracking using single camera with new extended Kalman filter. *Int. J. Intelligent Computing Cybernetics*, **1**: 379–397, 2008.
[24] A. Yilmaz, O. Javed, and M. Shah. Object tracking: a survey. *J. ACM Computing Surveys*, **38**, Article No. 13, 2006.

Junli Tao is a PhD student at Auckland University, New Zealand. She received her Master degree in 2010 from Hunan University at Hunan, China. Her research interests are in active pedestrian protection systems for diver assistance, described by keywords such as ego-motion analysis, pedestrian detection and tracking, and pedestrian intention prediction.

Dr. Reinhard Klette is a professor at the Computer Science Department at Auckland University, New Zealand. His professional interests are in computer vision (theory and applications) and in the design of geometric algorithms. He (co-)authored more than 350 peer-reviewed publications. He is the Editor-in-Chief of the Journal on Control Engineering and Technology.