A Precise Tracking Algorithm Based on Raw Detector Responses and a Physical Motion Model

Oliver Birbach Udo Frese

***Abstract*— We present a method to simultaneously track multiple objects which are subject to physical motion and can be evaluated through raw detector responses in video. Due to their two-staged design, popular tracking-by-detection approaches lack precision in the estimated trajectories due to detector inaccuracies, e.g., lighting, deformation or background clutter. Instead of separating the tasks of detection and tracking, we propose to integrate both in a single probabilistic objective function for determining the object states in a sequence. Both support each other accounting for detection inaccuracies and leading to a robust and precise single target tracker. Based on this, we extend it to multiple targets by solving the problem of determining trajectory limits and sorting out any multiple target ambiguities probabilistically. We apply our method to the task of tracking thrown balls with the goal of accurate trajectory prediction for the purpose of ball catching with a humanoid robot. Our results show improved tracking accuracy with respect to ground truth on average by around 17 %, which is dominated by increased accuracy at the beginning of the trajectory.**

I. Introduction

Having a system capable of robustly tracking multiple targets in image sequences is a prerequisite for many computer vision and robotic tasks. Usually, such a system consists of applying an object detector to images and linking the results to tracks, a task also known as data association. In such a bottom-up approach, the quality of the resulting trajectories is mainly governed by two factors. First, the reliability of the object detector regarding missed, inaccurate and false-alarm detections, and second, the ability of the ensuing tracking algorithm to correctly detect and deal with any errors propagated from the detector.

Because of this error propagation, current efforts in tracking reveal weaknesses when confronted with the case of tracking an object which is subject to physical motion. For example, vision-based approaches focus on appearance cues, e.g., HOG descriptors or color histograms, while neglecting motion characteristics. Besides the rough overall trajectory approximation, occasional errors in detection further lead to considerable local deviations (Fig. 1, left). On the other side, classical tracking approaches use detector peaks to estimate the parameters of a, e.g., constant velocity, motion model. While these give smooth trajectories, any detection error affects the whole trajectory substantially (Fig. 1, middle).

O. Birbach and U. Frese are with German Center for Artificial Intelligence (DFKI), Cyber-Physical Systems, 28359 Bremen, Germany {oliver.birbach,udo.frese}@dfki.de. U. Frese is also with University of Bremen.

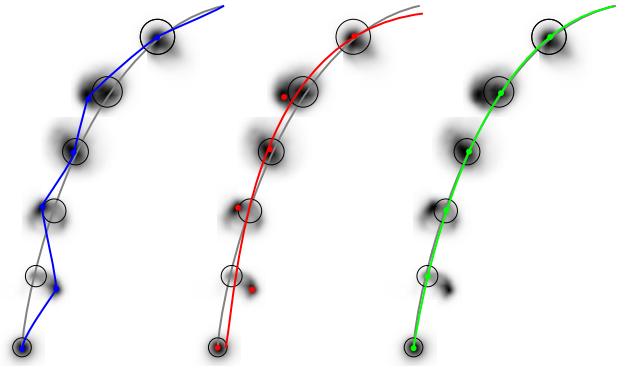

Fig. 1. Comparison of trajectories resulting from different tracking paradigms applied on raw detector responses at different time steps. Please note that the highest detector response (indicated as darker gray levels) is not necessarily on the actually observed trajectory (indicated as a gray-colored curve) due to, e.g., lighting variations. Appearance focused tracking **(left)** snaps into the detector's maxima ignoring the overall dynamic properties. The resulting trajectory is dominated by pairs of detections and loses local accuracy with wrong ones. Classical single target tracking **(middle)** fits a physical motion model to the detector's maximum responses. Again, wrong detections impair the quality of the trajectory but this time on a global level. We propose to integrate the tasks of detection and tracking with a physical motion model **(right)**. The trajectory is significantly improved as tracking and detection mutually support each other during estimation.

We argue that the lack of proper inclusion of the motion model guiding a response-based detector is the limiting factor. As motion provides substantial context regarding object locations over time, we suggest to leverage this motion evidence in a top-down fashion so that both previously separated tasks are supporting each other (Fig. 1, right).

In detail, we integrate both response-based detection and tracking using a physical motion model into a single probabilistic objective function for continuous trajectory estimation. Given these, probabilistically modeled trajectory boundaries and compatibility is determined by discrete optimization. Because of this thorough probabilistic formulation, we call the resulting tracker the *fully probabilistic multiple target tracker* (FPMTT). Please note, that we are not doing explicit data association (DA) on sets of discrete observations but work with raw detector responses evaluated in the image for each state. We only utilize DA for initialization and once a trajectory is started it is guided by the motion model where the continuity solely depends on response evidence.

This work is part of the effort in estimating and predicting multiple balls pitched towards a humanoid robot with the goal to catch each ball with one arm [1]. Up to now, we relied on a tracking-by-detection approach using a Multiple

Hypothesis Tracker (MHT) / Unscented Kalman Filter (UKF) fed by detector maxima [2]. Admittedly, the multiple target tracking problem is rather simple, but trajectory accuracy highly depends on the quality of detection and for a faithful treatment a full multiple target solution had to be integrated.

Our contributions are, *(a)*, a probabilistic framework for tracking multiple objects based on raw detector responses and a physical motion model extended by *(b)*, a way for extracting the trajectory's boundary from a sequence by using a modified maximum subarray method and, *(c)* a formulation of trajectory compatibility using the generalized independent set approach. Additionally, we give an instance (*(d)*) of this framework for the problem of tracking flying balls, and evaluate our approach regarding trajectory accuracy using ground truth.

The rest of the paper is structured as follows. We discuss related work in the next section. Our proposed method is introduced in Sec. III. The instance for the task of ball tracking is presented in Sec. IV with experiments in Sec. V.

II. Related Work

Besides the ongoing effort in developing sophisticated data association algorithms handling difficult detection situations [3]–[5], coupling detection and tracking/data association of multiple objects in image sequences became an active field of research, mostly in computer vision.

Leibe et al. [6] and later Ess et al. [7] were one of the first to model detection and trajectory estimation in a combined optimization as a quadratic Boolean problem (QBP). They multiply the prediction distribution of a Kalman Filter into the detectors response searching for the maximum of the product, but still use only that maximum, not the full response distribution to update the estimate.

Andriyenko et al. [8], [9] formulated multiple target tracking as a continuous energy minimization problem with a constant velocity motion model. However they still use an analytical distribution around detector maxima and not the original detector responses.

Wu et al. [10] couple both tasks in a single objective function, modeled through sparsity driven detection and network flow data association. Lagrange dual decomposition is used for optimization, but the state-space is discretized, so this approach probably does not scale up to, e.g., 3-D position and velocity as we have.

Recently, Collins [11] suggested to leverage kinematic motion in cases where appearance is similar and proposes a spline energy cost function to assess trajectory quality in challenging multiple target tracking scenarios.

The concept of using independent sets to determine the global hypothesis without violating any multiple target constraints has been subject of prior research. Papageorgiou and Salpukas [12] establish a maximum weight independent set (MWIS) formulation for solving the data association problem in traditional MHT. In computer vision, Brendel et al. [13] also use MWIS to guide formation and hierarchical concatenation of targets for tracking pedestrians in video.

Most of this work models analytical distributions around a set of (pedestrian) detections returned by a detector with the goal to robustly estimate trajectories and their interactions. This is reasonable, as motion of people is not reliably predictable and pedestrian detectors do not have pixel-level accuracy: robustness is favored instead of accuracy. Instead, we estimate the states of objects using detector responses directly, guided by a physical motion model as context. To our knowledge this is the first work to optimize the state of moving objects directly on image detector responses, instead of using local detector maxima as input.

III. Proposed Tracking Approach

We define our approach in a Bayesian sense with the goal of maximizing a likelihood, i.e. $\arg\max_x p(X = x|Z = z)$. Instead of using sets of measurements consisting of image coordinates computed from a preceding detection stage, we define Z to be the images themselves. Now, detection becomes part of the probabilistic optimization process and, guided by a motion model, is able to find out for itself whether or not the object is located at certain image coordinates. This is central to our contribution and becomes apparent when compared to the two-staged process, where the reduction of a detection to image coordinates causes a considerable information loss. By keeping images in memory and reevaluation at the corresponding image portions defined by the states, *all* available evidence from the images is used for tracking, greatly benefiting robustness and accuracy.

A. Model

In multiple target tracking, we want to find an unknown number of tracks n_a and, between the time of track starting $t_{\text{start}}^{(a)}$ and track ending $t_{\text{term}}^{(a)}$, each track's states $x_{t_{\text{start}}^{(a)}}^{(a)} \dots x_{t_{\text{term}}^{(a)}}^{(a)}$.

With this notation, we model $p(X = x|Z = z)$ as a product of probabilities where each of them handles a certain part of the overall problem. As common in these approaches, we write the function in convenient log-likelihood notation:

$$
\begin{aligned}
&p(X = x|Z = z) \propto \exp L(x), \text{ where } L(x) = \\
&\sum_{a=1,t=t_{\text{start}}^{(a)}}^{a=n_a,t=t_{\text{term}}^{(a)}} L_{\text{det}}\big(z_t, x_t^{(a)}\big) + \sum_{a=1,t=t_{\text{start}}^{(a)}}^{a=n_a,t=t_{\text{term}}^{(a)}-\delta t} L_{\text{dyn}}\big(x_{t+\delta t}^{(a)}, x_t^{(a)}\big) + \\
&\sum_{a=1}^{a=n_a} L_{\text{s\&t}}\big(t_{\text{start}}^{(a)}, t_{\text{term}}^{(a)}\big) + \sum_{\substack{a,a'=1,a>a' \\ t=\max(t_{\text{start}}^{(a)}, t_{\text{start}}^{(a')})}}^{\substack{a,a'=n_a, \\ t=\min(t_{\text{term}}^{(a)}, t_{\text{term}}^{(a')})}} L_{\text{exc}}\big(x_t^{(a)}, x_t^{(a')}\big)
\end{aligned}
\tag{1}
$$

and where

- $L_{\text{det}}\big(z_t, x_t^{(a)}\big)$ indicates support of object $x_t^{(a)}$ in image z_t,
- $L_{\text{dyn}}\big(x_{t+\delta t}^{(a)}, x_t^{(a)}\big)$ indicates the likelihood that an object transitions from $x_t^{(a)}$ to $x_{t+\delta t}^{(a)}$,
- $L_{\text{s\&t}}\big(t_{\text{start}}^{(a)}, t_{\text{term}}^{(a)}\big)$ is a prior indicating how likely an object emerges at $t_{\text{start}}^{(a)}$ and disappears at $t_{\text{term}}^{(a)}$ and
- $L_{\text{exc}}\big(x_t^{(a)}, x_t^{(a')}\big)$ indicates how likely objects are subject to occlusion.

B. Algorithm

As can be seen from the involved parameters, inference requires to solve a hybrid optimization problem. Estimation of states $x_t^{(a)}$ along a track is continuous, while the set of tracks and determining $t_{\text{start}}^{(a)}, t_{\text{term}}^{(a)}$, due to nature of image recording, are discrete. Our proposed algorithm (Alg. 1) is therefore constructed around three subproblems of optimizing (1) regarding different variables.

Trajectory Estimation: We treat trajectory estimation per track, while holding t_{start} and t_{term} fixed. Being one major contribution of this work, we propose to estimate all states along a trajectory simultaneously using both raw detector responses and a motion model:

$$\underset{x_{t_{\text{start}}} \dots x_{t_{\text{term}}}}{\arg\max} \sum_{t=t_{\text{start}}^{(a)}}^{t=t_{\text{term}}^{(a)}} L_{\text{det}}\big(z_t, x_t\big) + \sum_{t=t_{\text{start}}^{(a)}}^{t=t_{\text{term}}^{(a)}-\delta t} L_{\text{dyn}}\big(x_{t+\delta t}, x_t\big) \tag{2}$$

The likelihood of observing an object L_{det} is provided by an evaluation function which looks at that position in the image z_t that corresponds to x_t and assesses how much it looks like the object. Actual modeling of the likelihood might depend on the type of object and intended task. Please c.f. Sec. IV-A for our likelihood on radial image contrast for circular shapes as an example.

L_{dyn} is modeled as the quadratic error between the mapping of a state x_t to time $t + \delta t$ using dynamic function g and state $x_{t+\delta t}$ while considering noise $\sim \mathcal{N}(0, \Sigma_g)$:

$$L_{\text{dyn}}\big(x_{t+\delta t}, x_t\big) = -\|x_{t+\delta t} - g(x_t)\|^2_{\Sigma_g} \tag{3}$$

Unfortunately, solving Eq. 2 is not trivial. The tight coupling of states due to L_{dyn} leads to bad conditioning, i.e. some dimensions are stiff while others are not, requiring the use of a preconditioner for efficient optimization. Established approaches for this kind of nonlinear optimization problems are preconditioned conjugate gradient methods such as PNCG [14].

Track Limits: Determining the time of start t_{start} and termination t_{term} of a hypothetical track is also done per track. Given the sequence of fixed states, e.g., from a previous time step (and probably extended using the motion model g), we are interested in finding $\arg\max_{t_{\text{start}}, t_{\text{term}}}$ of

$$\begin{aligned} L\big(\{(t_{\text{start}}, t_{\text{term}}, x_{t_{\text{start}}} \dots x_{t_{\text{term}}})\}\big) = \sum_{t=t_{\text{start}}^{(a)}}^{t=t_{\text{term}}^{(a)}} L_{\text{det}}\big(z_t, x_t\big) + \\ \sum_{t=t_{\text{start}}^{(a)}}^{t=t_{\text{term}}^{(a)}-\delta t} L_{\text{dyn}}\big(x_{t+\delta t}, x_t\big) + L_{\text{s\&t}}\big(t_{\text{start}}, t_{\text{term}}\big) \end{aligned} \tag{4}$$

Here, likelihoods of track appearance and termination are modeled as

$$L_{\text{s\&t}}\big(t_{\text{start}}, t_{\text{term}}\big) = \log p_{\text{start}} + \begin{cases} \log p_{\text{term}} & t_{\text{term}} < t_{\text{now}} \\ 0 & t_{\text{term}} = t_{\text{now}} \end{cases} \tag{5}$$

where p_{start} and p_{end} denote the prior probability of target appearance and termination. The casted problem can be interpreted as a maximum subarray problem in a sequence which can be efficiently solved using Kadane's algorithm [15] in linear time, even when modified to include $\log p_{\text{start}}$ and $\log p_{\text{term}}$.

Algorithm 1 Fully Probabilistic Multiple Target Tracker

Input: Set of prior tracks A
Output: Most likely set of tracks A'
Set of posterior tracks A

- Insert initial trajectories as new tracks into A, mark them to do only trajectory estimation

for $x^{(a)} \in A$ **do**

- Extend tracks according to dynamic model g
- Determine track boundaries
  Solve $\arg\max_{t_{\text{start}}^{(a)}, t_{\text{term}}^{(a)}}$ of Eq. 4
- Estimate trajectory between boundaries
  Solve $\arg\max_{x_{t_{\text{start}}^{(a)}} \dots x_{t_{\text{term}}^{(a)}}}$ in Eq. 2

end

- Ensure mutual exclusivity by stating GIS problem
  Solve $A' \leftarrow \arg\max_{A \subset \{1..n\}}$ in Eq. 7
- Prune tracks in A with low likelihood

Mutual Exclusion: On the one hand, multiple similar tracks are generated by the same object during tracking. On the other hand, real objects may naturally occlude each other. We want to define an exclusion mechanism in a way that prevents the first and allows the latter.

The idea is to state a prior that objects occlude with a probability p_O, i.e. sometimes but not too often. Two states are assigned this prior as a penalty term when their projections into the image overlap.

$$L_{\text{exc}}\big(x_t^{(a)}, x_t^{(a')}\big) = \begin{cases} \log p_O & \text{if } x_t^{(a)} \text{ and } x_t^{(a')} \text{ overlap} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

We then extend this to the set of all tracks and model track interaction as a Generalized Independent Set (GIS) problem [16] where each track's likelihood (Eq. 4) contributes to its existence while overlap between two tracks penalizes both. The goal is then to obtain the optimal subset of tracks (while holding all other parameters fixed):

$$\begin{aligned} \hat{x} &= \arg \max_{A \subset \{1..n\}} L(\{x^{(a)} | a \in A\}) \\ &= \arg \max_{A \subset \{1..n\}} \sum_{a \in A} L(\{x^{(a)}\}) \\ &\quad + \sum_{a,a' \in A, a<a'} \sum_{t=\max(t_{\text{start}}^{(a)}, t_{\text{start}}^{(a')})}^{t=\min(t_{\text{term}}^{(a)}, t_{\text{term}}^{(a')})} L_{\text{exc}}\big(x_t^{(a)}, x_t^{(a')}\big) \end{aligned} \tag{7}$$

The resulting subset is the set of tracks likely to be existent given the evidence in the image. Please note, that the GIS problem is NP-complete and approximation has to be performed for large tracking scenarios.

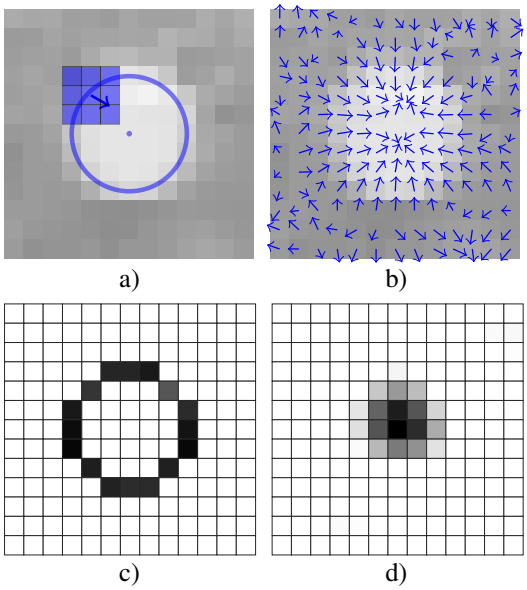

Fig. 2. Concept of the used circle detector [2]: **a)** Evaluation of gradient vector at a circle pixel to measure radial contrast, **b)** vector image after filter application, **c)** response (Eq. 8) at pixels along a fixed size circle, **d)** circle response (Eq. 9) for $r = 3$ and different center x_c, y_c.

Track Management: The set of tracks A in Alg. 1 needs to be maintained over time, i.e. new tracks must be added as well as terminated and unlikely tracks removed to prevent growing computation time. Both has been realized in an application specific way (Sec. IV-C). In general, pruning should make use of the GIS solution to determine which tracks to discard.

IV. Ball Tracking Instance

We now describe our instance of the proposed algorithm for the task of tracking multiple balls in image sequences. This includes the detection likelihood L_{det}, the dynamic function g of L_{dyn} as well as several implementation details for track management.

A. Observation Likelihood

Circle Response: We detect balls in gray scale images by their appearance as circles. For increased robustness in low contrast areas, we use an enhanced Sobel gradient filter C which normalizes for local image variance (see Fig. 2 a-b) and [2] for details). Intuitively speaking, instead of indicating gradient intensity as classical gradient filters do, the output of C indicates how perturbed a linear gradient is. Here, a value of 1 points to an unperturbed linear gradient where gradually lower values indicate a deviation from that.

The radial contrast of a local image can then be defined as a function of a point (x, y) along the circle and radial direction α (see Fig. 2c))

$$R(x, y, \alpha) = \left(\begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \cdot C(x, y) \right)^2 \tag{8}$$

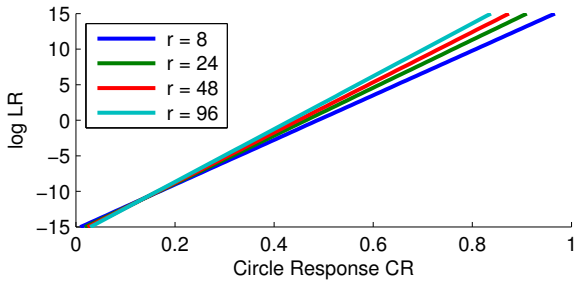

Fig. 3. Log of likelihood ratio as defined in Eq. 10 as a function of circle response (Eq. 9) for various circle radii. Please note how larger circles are favored at the same response value. This is due to the fact that it is more likely that the response of a small circle is incidentally high than the one from a large circle.

and the overall response for a circle at x_c, y_c, r is obtained by integrating along the circle (Fig. 2d):

$$CR(x_c, y_c, r) = \frac{1}{2\pi} \int_{\alpha=0}^{2\pi} R(x_c + r\cos\alpha, y_c + r\sin\alpha, \alpha) d\alpha \tag{9}$$

Likelihood Model: To use (9) as L_{det}, it needs to be converted to a log-likelihood. We obtain this by considering the ratio between the probability that a certain response and radius combination is generated by an actual ball and the probability that it is generated by the background [17]:

$$LR(x_c, y_c, r) = \frac{P_{ball}(CR(x_c, y_c, r))}{P_{bg}(CR(x_c, y_c, r), r)} \tag{10}$$

Here, likelihood ratios indicating a ball are generated when the statistics for a ball are better matched than these of the background. Similarly, when the ball model cannot explain a response better than the background model, low ratios are generated. In detail, we use the heavier-tailed Laplace distribution to prevent unreasonable probabilities at extreme responses. To better capture the background distribution at different circles sizes, the radius is included in the model. Furthermore, we approximate the distribution slightly to realize a linear function for the log likelihood-ratio (Fig. 3).

$$P_{ball}(cr) = \frac{1}{2b} \exp\left(-\frac{\mu_{ball} - cr}{b_{ball}}\right) \tag{11}$$

$$P_{bg}(cr, r) = \frac{1}{2b\, r^{\gamma}} \exp\left(-\frac{cr - \mu_{bg}}{b_{bg}\, r^{\gamma}}\right) \tag{12}$$

Distribution parameters haven been obtained by closed form maximum likelihood fitting to histogram data. Histograms were created from responses of patches containing a ball and image sequences containing random scenes.

The mapping of a ball state x_{ball} to a circle (x_c, y_c, r) is achieved through projection h (including pinhole-model camera calibration) leading to $L_{\text{det}}(x_{\text{ball}}) = \log LR(h(x_{\text{ball}}))$.

B. Physical Motion Model

The dynamic of the ball during flight is modeled by classical mechanics including gravitation $\mathbf{g}$ and air drag α.

$$\dot{\mathbf{x}} = \mathbf{v}, \quad \dot{\mathbf{v}} = \mathbf{g} - \alpha \cdot |\mathbf{v}| \cdot \mathbf{v} \tag{13}$$

The dynamic function g is then the Euler integration of Eq. 13 over δt.

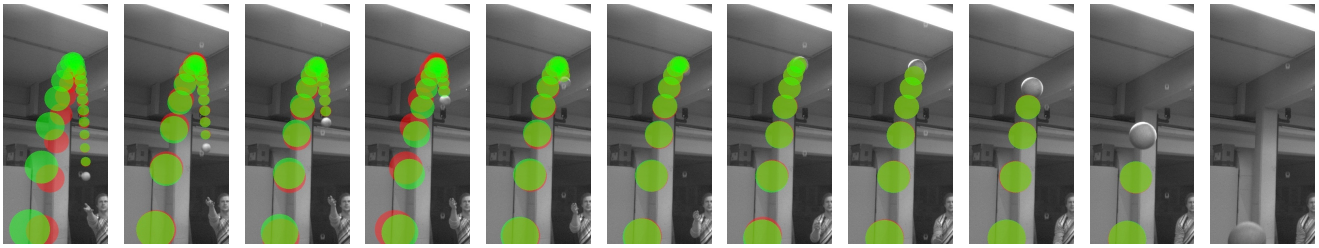

Fig. 4. Sequence of clipped images, recorded during throwing a ball towards the robot, showing the predicted trajectory over time as computed by MHT/UKF **(red)** and our approach **(green)**. It can be seen that the green trajectory is already quite accurate at the beginning while the red trajectory suffers from early detection inaccuracies and recovers over time to reach the same final accuracy. This impressively illustrates the increased performance at an early tracking stage. Please see the supplemental video for the full sequence.

C. Implementation Details

For the task of ball catching, the robot is equipped with stereo cameras and an inertial measurement unit (IMU). The latter provides the camera's orientation regarding gravity for proper tracking and prediction of the ball's trajectory.

Both cameras provide valuable information through implicit triangulation, so L_{det} becomes $L_{\text{det}}^{\text{left}} + L_{\text{det}}^{\text{right}}$. We keep a history of the processed vector images in memory for 1 s so that ball trajectories are contained in their entirety. For each state and image, we buffer a $32 \times 32 \times 32$ px volume of the log-likelihood ratio in memory and use the tricubic approach by Lekien and Marsden [18] for interpolation. This ensures that L_{det} is smooth at subpixel-level which is required for proper operation of the optimizer.

Trajectory estimation is performed using PNCG where the preconditioner matrix should be the inverse of the Hessian of Eq. 2. To obtain this matrix we approximate L_{det} as a Gaussian with a vague uncertainty making Eq. 2 quadratic. From there the Hessian can be computed straightforwardly. Please note that we are using this only for the construction of the preconditioner matrix, which only affects the rate of convergence, not the optimization result.

In most approaches, track initialization is a special case handled in an application specific way. In our approach, trajectory estimation is dependent on good initial tracks as convergence with states far off the local maxima is not guaranteed. Therefore, we used the existing MHT/UKF and whenever a track started there, we also initialized a new track in the FPMTT.

Trajectories are extended at most one step into the past and two steps into the future in every iteration. The former allows to include possible measurements missed by track initialization. This greatly benefits early tracking precision, when the track initialization mechanism missed measurements. The latter allows the tracker to revise the decision of a track having ended in light of new evidence.

As the number of tracks we need to handle (two for the case of two-handed catching and a couple more for evaluation) is moderate, the GIS problem is solved exhaustively in combination with smart pruning in each iteration. While solving, we maintain a list of the k-best subsets of tracks and keep only the included ones up to a threshold.

V. Experiments

As the joint optimization for trajectory estimation is the key of our approach, we are mainly interested in single-target performance and defer multiple target evaluation to future work.

To validate our algorithm we compare its prediction to ground truth (possible catch point) obtained by a 3-D tracking system. We chose to compare this to the state's *prediction* using the physical motion model, as this corresponds to the intended application of catching thrown balls. This sensitive task requires accurate predictions for reliable catching and its success depends on early tracking quality.

An image sequence indicating the aforementioned benefits of our approach compared to MHT/UKF is given in Fig. 4. Guided by the physical motion model, global evaluation of the trajectory allows to lock in at the real circle evidence, especially in the early stage of tracking.

Figure 5 quantifies this as the average error ratio over a set of 48 trajectories from different throwing sessions. As shown in the individual plots, FPMTT outperforms our previous approach most of the times and on average by 16.5 %.

Detector inaccuracies for ball detections (see Fig. 6) are mainly caused by low contrast due to varying lighting or background interaction. Circles determined by detector maxima share some part of radial contrast of the ball's projection resulting in a smaller circle detection. Up to our inspection, our approach is able to resolve almost all of these and lock in on the partial circular shape of the projected ball.

Due to focus on proper convergence, real time performance is not achieved yet, but intended for the future. Current average computation time is a multiple of the allowed time per frame (single core, no high level optimizations).

VI. Conclusion

In this paper we presented a new algorithm, called the fully probabilistic multiple target tracker, for tracking objects which are subject to physical motion and can be evaluated through raw detector responses in video sequences. The algorithm breaks down to solving the three subproblems of trajectory estimation, determining the trajectory boundaries and handling mutual exclusion for which we proposed efficient solutions for the task of ball tracking.

After evaluation on recorded data sets, we want to achieve real time performance and integrate our algorithm into the actual robotic ball catching framework. Here, we are especially interested how much the planning stage in the robotic catching system benefits from the improved prediction accuracy. It is expected that it leads to smoother and more visually appealing arm trajectories.

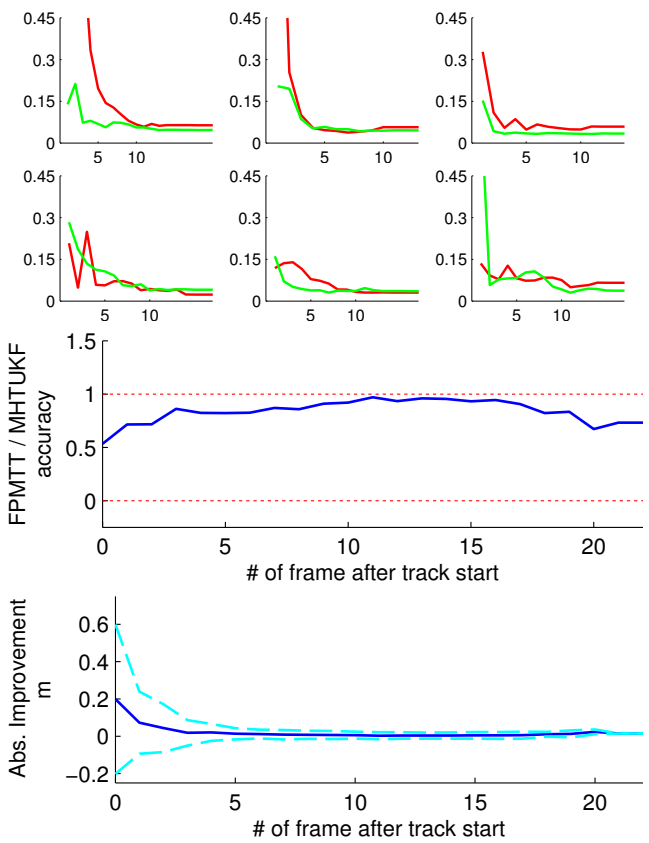

Fig. 5. **(1st row)** Excerpt of typical prediction accuracy results (in m) after track start. FPMTT (green line) is able to outperform MHT/UKF (red line) by a great margin. **(2nd row)** Examples where FPMTT is outperformed by MHT/UKF but achieves still reasonable performance. **(3rd row)** Geometric mean of the error ratio between FPMTT and MHT/UKF at each time step after track start. As expected, in the beginning the error can be reduced almost 50 % until it both perform the same at frame 10. The also slightly improved accuracy at the end is likely due to better treatment of circles at the image's border (ball leaves image) of our approach. **(4th row)** Average prediction improvement after track start of FPMTT (solid line) and 1σ deviation (dashed line) to give and indication what range of improvement in the beginning we can expect for our accuracy dependent task.

Leaving the field of ball tracking we want to generalize our approach to different tracking applications in image sequences. In detail, we want to investigate how more complex geometrical appearance models (e.g., from 3-D models) can be integrated. Furthermore, we are interested in finding out how loose the motion model can be while tracking with our method is still feasible.

VII. Acknowledgments

This work was supported under DFG grant FR2620/1-1 as well as grant SFB TR/8 Spatial Cognition.

References

[1] B. Bäuml, O. Birbach, T. Wimböck, U. Frese, A. Dietrich, and G. Hirzinger, "Catching flying balls with a mobile humanoid: System overview and design considerations," in *Proc. of the IEEE-RAS Int'l Conf. on Humanoid Robots*, 2011, pp. 513–520.

[2] O. Birbach, U. Frese, and B. Bäuml, "Realtime perception for catching a flying ball with a mobile humanoid," in *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, 2011, pp. 5955–5962.

[3] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1918, 2005.

[4] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

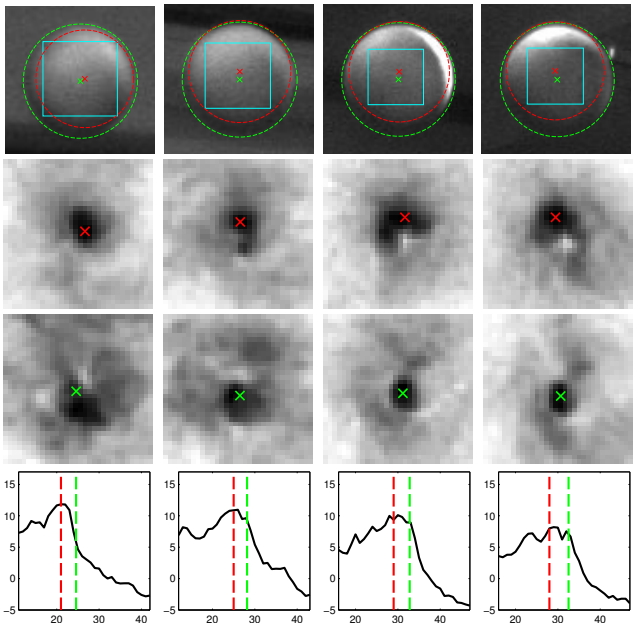

Fig. 6. Four examples of detector inaccuracies and their handling in our approach. **(1st row)** Image excerpt around the ball where a red circle denotes the detector maxima and a green one shows the circle determined by our approach. Plot of LR volume (bounding box shown above) at radius of detector maxima **(2nd row)** and volume of found circle radius after trajectory estimation **(3rd row)**. Orthogonal plot showing LR as a function of ball radius for both circles **(4th row)**.

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

[6] B. Leibe, K. Schindler, and L. V. Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *Proc. of the Int'l Conf. on Computer Vision*, 2007.

[7] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1831–1846, 2009.

[8] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011, pp. 1265–1272.

[9] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 1926–1933.

[10] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 1948–1955.

[11] R. T. Collins, "Multitarget data association with higher-order motion models," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 1744–1751.

[12] D. Papageorgiou and M. Salpukas, "The maximum weight independent set problem for data association in multiple hypothesis tracking," *Optimization and Cooperative Control Strategies*, pp. 235–255, 2009.

[13] W. Brendel, M. Amer, and S. Todorovic, "Multiobject tracking as maximum weight independent set," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011, pp. 1273–1280.

[14] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," School of Computer Science, Carnegie Mellon University, Tech. Rep., 1994.

[15] J. Bentley, "Programming pearls: Algorithm design techniques," *Comm. of the ACM*, vol. 27, no. 9, pp. 865–873, 1984.

[16] D. Hochbaum, "Instant recognition of polynomial time solvability, half integrality, and 2-approximations," in *Approximation Algorithms for Combinatorial Optimization*, 2000, pp. 379–405.

[17] H. Sidenbladh and M. J. Black, "Learning image statistics for Bayesian tracking," in *Proc. Int'l Conf. on Computer Vision*, 2001, pp. 709–716.

[18] F. Lekien and J. E. Marsden, "Tricubic interpolation in three dimensions," *Int'l J. Numer. Methods Engrg.*, vol. 63, pp. 455–471, 2005.