Adaptive Human Motion Prediction using Multiple Model Approaches

Markus Joppich, Dominik Rausch, Torsten Kuhlen

Virtual Reality Group
Center for Computing & Communication
RWTH Aachen University
Seffenter Weg 23, 52074 Aachen
E-Mail: info@vr.rwth-aachen.de

Abstract: A common problem in Virtual Reality is latency. Especially for head tracking, latency can lead to a lower immersion. Prediction can be used to reduce the effect of latency. However, for good results the prediction process has to be reliably fast and accurate. Human motion is not homogeneous and humans often tend to change the way they move. Prediction models can be designed for these special motion types. To combine the special models, a multiple model approach is presented. It constantly evaluates the quality of the different specialized motion prediction and adjusts the set of motion models. We propose two variants, and compare them to a reference prediction algorithm.

Keywords: Adaptive Prediction; Latency; Multiple Model Estimation

1 Introduction

One of the key aspects of Virtual Reality (VR) is the reproduction of highly immersive virtual scenes. For this, it is not only important that the output is realistic, but also that it adopts to the user's interactions. For example, when moving the head, the displayed images have to be adjusted to maintain a correct perspective. One common problem of interactive environments is latency, i.e. the delay between input and the reaction in the output.

For a user-centered projection, a high latency can cause motion sickness or make the user move unnaturally [SMK98]. When rendering audio using binaural synthesis using dynamic cross-talk cancellation [LSVA07], the precise position of the listeners ears is required. If the user moves, the filter computations assume a wrong ear position due to latency, resulting in a loss of quality of the sound reproduction.

There are several different causes of latency, for example the update frequency of the tracking system, network delay, rendering speed, or display refresh rate. It is often not possible to easily reduce the latency, e.g. because better hardware is not available or too expensive. Thus, latency can only be reduced to a certain level. For example, the CAVE at RWTH Aachen University has a visual end-to-end latency of 70 to $100ms$.

When the possibilities of a direct reduction of latency are exceeded, prediction can help to reduce the impact of latency. Prediction uses information from past measures to estimate

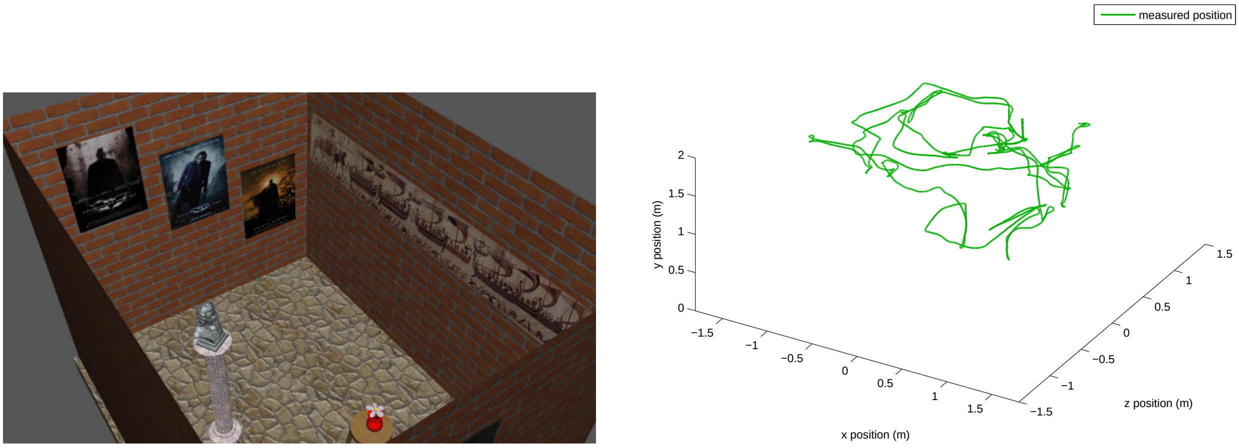

Figure 1: Layout of the virtual scene (left) and a recorded head trajectory(right).

a future state. These future states can then, for example, be used to render images or sound based on the expected head position. However, especially for long prediction times, the prediction results may be inexact, possibly causing even larger deviations or sudden jumps (jitter). Thus, it is important to find reliable and accurate prediction methods and models.

Several methods for predicting human motion in VR have been proposed. Many of these strive for an abstract model that can predict just about any motion. However, when moving around a virtual scene, e.g. as in figure 1, humans typically perform different actions. These can often be categorized as a set of different *motions types*, e.g. walking straight ahead, looking around, or kneeling down. For these specific motion types, optimized prediction models outperforming general approaches might be found. However, since humans tend to change their motion over time, the prediction has to be *adaptive* by adjusting tracking parameters or switching motion models.

In this paper, we will present methods for multiple model adaptive estimation for optical head-tracking in VR. For this, we will present specialized position prediction models for different movement types using Kalman Filters. These are combined into two different multiple model prediction methods. The remainder of this paper is structured as follows. We will first discuss related work in section 2, followed by a description of basic prediction methods in section 2.1. The specialized motion models and the multiple model approaches are presented in section 3. Section 4 will present results of the proposed methods compared to a reference adaptive Double Exponential Smoothing (aDES) implementation, followed by a conclusion and outlook (section 5).

2 Related Work

Work on prediction algorithms, also in the context of VR, already has a long history. Especially because of head-mounted displays, a lot of work focusses on prediction of head orientation using inert sensors [vRM05].

Probably the most common method for prediction is the Kalman Filter [Kal60], which has been used for several applications in VR (see [Wel09] for a history). Other approaches make use of Particle Filters [ADMR03]. Another popular class of methods use Double Exponential Smoothing (DES), as e.g. presented in [LaV03]. This algorithm is simple compared to Kalman Filters (KFs), and thus can be computed a lot faster, but still provides high accuracy. Due to its simplicity and computational efficiency, DES gained attention, e.g. for jitter reduction [CK13] or locomotion prediction [NK13].

Several prediction algorithms for orientation have been systematically evaluated [vRM05]. The results show that the Extended KF, Unscented KF and Particle Filter based prediction methods all deliver a similar accuracy for orientation predictions of less than $80ms$. Another work compares the use of quaternions vs. *delta-quaternions* for orientation prediction using a KF [HM09]. The authors conclude that delta-quaternions can enhance the prediction quality.

While general prediction is rather common in VR, adaptive prediction approaches are less frequent. An adaptive variant of the DES has been proposed [Ass09], which dynamically adjusts the parameters of the prediction to match the current motion. Alternatively, one can use several different specialized models and adaptively choose the best (or a combination of the best) by a method called a Multiple Models Adaptive Estimator (MMAE) [KM97]. Chai et al. [CHVN99] have presented an implementation of an adaptive estimator, which performs slightly better than the best non-adaptive estimator, but at higher computational costs. It is stressed that for a general system with no a-priori information it is not possible to know the best-matching non-adaptive model in advance.

The main challenge when using MMAE for prediction of human motion is the determination and combination of suitable specialized models. In this paper, two approaches to solve this problem will be presented.

2.1 Basic Prediction Techniques

This section briefly introduces prediction concepts using non-adaptive prediction methods. While a Double Exponential Smoothing (DES) implementation is used as reference, Kalman Filters (KFs) will be used to model specific motion models in section 3.

Exponential Smoothing Exponential smoothing algorithms have been developed primarily to smooth noisy data series. They are also a viable prediction method, especially in economic forecasts [Gar85]. Double Exponential Smoothing (DES) is a further improved exponential smoothing scheme and is proposed as an alternative to Kalman Filtering [LaV03]. In contrast to single exponential smoothing it models two separate trends: a current and a general one.

The performance of the DES depends on a good choice of the smoothing factor. While this factor is often kept constant, it can also be adapted interactively [Ass09].

This adaptive algorithm will be used as reference implementation with which we will compare our results.

Kalman Filter In this section, we will briefly explain the fundamentals of Kalman Filters. For a more detailed discussion, see [BH12].

The KF is a filtering technique used to calculate estimates of unknown variables based on a series of observed measures. For each measure, a prediction step and a correction step are performed. The calculated estimate often tends to be more accurate than a single observation, as noise is filtered out. A main assumption for KFs is that there is an underlying linear dynamics model and all error terms and measurements have a Gaussian distribution. Thus KFs are often used to filter noise from observations. Human motion has been found to be well linearizable, and thus KFs can be applied for prediction of human motion [KB06]. The Kalman Filter assumes a random process which can be estimated at discrete time-steps as $x_{k+1} = \phi_k^{k+1} \cdot x_k + w_k$, where x_i is the state vector at time-step i, ϕ_k^{k+1} is the state transition matrix transforming the state x_k into x_{k+1}, and w_k is the process noise covariance. The random process can be observed at discrete points in time by $z_k = H_k \cdot x_k + v_k$ where H_k denotes the measurement correction matrix and v_k is the measurement noise covariance. ϕ_k^{k+1}, H_k, w_k and v_k must be given and can be determined from the dynamics model and from experiments.

In the first step from k to $k+1$ the *a posteriori* state x_k^+ at time t_k is transformed into the *a priori* state x_{k+1}^-. The second step is the correction step which corrects the *a priori* state at time k using the measurement from time k. In the prediction step from k to $k+1$ the a priori state x_{k+1}^- is calculated under the assumption of a noise-free dynamics model: $x_k^- = \Phi_{k-1}^k \cdot x_{k-1}^-$. The correction step calculates the *a posteriori* state from the a priori state using the most recent observation.

For further prediction, the N-step prediction technique can be used. The state transition matrix Φ_k^{k+N} is specific for each model. The predicted state vector is determined by $x^- (k + N \mid k) = \Phi_k^{k+N} \cdot x (k \mid k)$.

3 Prediction Methods

This section will first introduce specific motion models for human motion. These will then be combined to form multiple model approaches. Many parameters of the presented models and the covariance matrices of the Kalman filters depend on the tracking system as well as other system-specific properties (e.g. limited walking speed in a CAVE). They cannot necessarily be applied to predictive tracking with a different environment and hardware. Thus, we will omit specific values, and only present the general approach of the models and the estimators. The presented models are designed for single axis position prediction. For further implementation details, see [Jop12].

3.1 Dynamics Models and Kalman Filter Parameters

The KF requires a dynamics model which should be chosen according to the desired state transition. For performance reasons, the state vector should be kept small. A position-velocity-acceleration dynamics model has been chosen [KB06]. We recorded example trajectories from users exploring a virtual scene (see section 4), and evaluated these to determine different types of motions and to design individual, specialized motion models. The more important models that have been developed are: KF-based motion models for fast, slow and robust linear motion, parabolic motion, constant position, constant velocity change and exponential and linear change-to-velocity motion have been developed. Here, we will only describe the linear and the exponential change motion models, since these proved to be versatile and most suited for the multiple model estimation approach.

For multiple model approaches, all models receive the same input tracking states. While KFs inherently smooth the input, this is not always sufficient because velocity and acceleration have to be calculated numerically from the input positions, pronouncing errors. Thus, we decided to use additional smoothing in the form of a Kalman Smoother [SVL07].

Linear Motion Model *Linear motion models* describe a movement process during which velocity stays almost constant. It has been observed that a subdivision into fast and slow linear motion provides better results than just using a single class. Observations have shown that when moving quickly, the normal linear motion model slightly underestimates the actual position. Similarly, for slow linear movement it predicts a position that is too far ahead. Thus, the slow and fast motion models correct for these over- and underestimation by not using the actual measured velocity, but an adjusted one.

Exponential Change-to-Velocity Model Accelerating human motion would classically be modelled using a KF with a constant acceleration, which corresponds to the linear change-to-velocity motion model. However, using an acceleration derived from the input positions introduces a high jitter or requires strong smoothing that would introduce a delay. Additionally, our recordings showed that the acceleration does not stay linear during the process.

Thus, we propose the use of an *exponential change-to-velocity motion model*, which accelerates from one velocity to another in a given time. It exponentially interpolates between the initial velocity and a target velocity at each step. This model is *general but specific* because it can easily be adapted to specific cases by changing the parameters *target velocity* and *target interval*, i.e. the time after which the target velocity is reached. While this provides flexibility, it is important to choose the correct parameters to match the current motion model. This parameter determination will be performed by the MMAE.

3.2 Adaptive Estimation Using Multiple Models

The previously presented motion prediction models have been designed for specific cases of human motion. While tracking a human's head, the type of motion will frequently change, e.g. when one accelerates, turns around, or shifts his torso to look around a stationary object. In these cases, it is important to select the correct motion model at the correct time, which is a difficult problem. For this, a Multiple Models Adaptive Estimator (MMAE) is used to select or combine the matching specific models out of all possible models. The multiple model approach can use several instances of a given model (e.g. with different parameters). These instances are called filters.

The MMAE is based on the theoretic assumption that there is one true model and thus for each state a perfect estimate can be calculated [Mar06]. However, this approach is not feasible for interactive prediction because all possible filters and filter selection histories have to be examined. Instead, a more practical solution has been used by Chai et al. [CHVN99]. The selection process chooses one filter after every k-th frame, based on the filters' error over these k frames. This approach always uses the result of a single filter, thus discontinuities arise when a new filter is selected. Furthermore, the window length introduces a delay during which the estimator cannot react to changes.

In our case, several filters run simultaneously, and new filters are created or pruned at runtime. From these filters, we compute the predicted position as a weighted sum of each filter's result $\hat{x}(t_{k+N}|t_k) = \sum_{Filter\ i} \omega_i(t_k) \cdot \hat{x}_i(t_{k+N}|t_k)$, where $\omega_i(t_k)$ is a normalized error criterion. Our tests showed that the most suitable criterion is the inverse absolute error of the most current measurement and the filter's corresponding prediction.

3.2.1 Multiple Models Adaptive Estimator

In our approach, all filters contribute to the final prediction based on their current error. However, the measure of the current error only describes a momentarily error measure. Thus, a filter that coincidently yields a good match, but generally shows a wrong trend, would gain too much influence. While this can be mitigated by using a windowed error measure [CHVN99], this introduces the already mentioned problems. Instead, we decided to use a two-level method. For combining models, the current error representing short-term changes is used. Additionally, filters are dynamically created and removed according to the general trend of the trajectory. This ensures that only filters matching the current trend, i.e. the medium-term development of the user's movement, are regarded for the weighted sum.

For this purpose, methods are required to determine which active filters should be pruned and which new filters should be created. To determine if a filter should be deleted, error thresholds are checked. If a filter shows a high error, or a medium error over multiple frames, it is removed from the active set. For pruning, this simple approach was sufficient, but for creating new filters, a more complex heuristic is required. New filters are created whenever the current error exceeds a low error threshold. While it would be possible to create new filters at each time-step, this would introduce a large number of very similar filters, and would also increase the computational complexity.

To create new filters, a set of filters and their parameters have to be determined based on the current trajectory. To achieve this, some filter parameters are determined based on the current movement, while others are sampled over a meaningful interval. For example, for the *exponential change-to-velocity model*, the target velocity is sampled around the current velocity. Three target intervals were found to be sufficient to cover most cases: 8, 12, and 18 frames (i.e. 133, 200, and 300ms). Filters are created for all models listed in section 3.1.

Due to the large number of models and parameter combinations, a lot of filters are created which are valid at the current time-step, but do not match the general movement trend. To overcome this problem, all newly created filters are initialised for a point in the past (which has been chosen as 6 samples for a 4 samples prediction). Thus, the filters have a history allowing an immediate initialisation and evaluation at the current time to check their error. If a filter's prediction of these past measures diverges too much, it is immediately deleted. This reduces the number of actually used filters to a reasonable amount.

The error criteria used for pruning are not single thresholds, but always check for short-term high thresholds and for medium-scale increases of the error above a lower threshold. The values of the threshold are adjustable and depend on the used hardware as well as the prediction interval.

By combining the dynamic control of the filter set and the weighted sum, it is possible to check both the short-term and the medium-term error. This achieves better results than using just a single option. To conclude, choosing a filter creation heuristic according to these criteria allows an adaptive position prediction using a multiple model approach. Non-matching filters get pruned soon enough, and only filters matching the current situation are created.

3.2.2 Dual Model Adaptive Estimator

From observations of the MMAE, we noticed that despite the dynamic creation and pruning of filters, it often reacted too promptly to small changes in motion. Thus often a curve is predicted, when the actual movement only showed a small deviation from the linear case. However, if one would react too late, the predictor could miss the start of the curve, and would then cause larger errors. Thus, it is difficult to trade the delay with which the MMAE reacts to changes against possibly induced errors.

Another observation is that most of the time, human motion is mostly linear, with only short periods of acceleration. Furthermore, in critical areas such as the apex of a turn, humans typically move rather slow, so that the error induced by latency is lower than for fast linear movement.

These observations motivated the design of the Dual Model Adaptive Estimator (DMAE), a multiple model approach with only two different filters: the fast and slow linear motion models. Depending on the current velocity either the fast or slow linear motion filter is chosen. This simplified filter selection and reduced filter count reduces the computational complexity. It is also easier to implement, and fewer parameters have to be matched to the hardware.

By using only the linear motion models, the dominant linear movement periods are well predicted, while in curved regions the apex is typically overshot, and small details are lost. Due to the mentioned observations, this trade-off seems reasonable.

4 Evaluation

To evaluate the methods presented in section 3, we performed tests on actual tracking of users exploring a virtual scene. For this, we used two main measures for the quality: Mean Absolute Error (MAE) and *jitter*. *Jitter* is measured as the sum of direction changes in velocity, normalized by the observed changes of velocity from measurements. A jitter value close to one expresses that the prediction does not suffer from more jitter than the original tracking signal. A measure larger than one indicates more fluctuations, while a measure smaller than one indicates a smoother – although not necessarily more accurate – prediction. The error is compared to an error that would occur with no prediction. Here, the error is induced only by latency, which is assumed to be the prediction interval of four samples.

The scene and the position samples used as test input for the prediction are shown in figure 1. Tasks in this scene included looking at posters, moving around the room as well as taking a closer look at a statue. The tracking input is recorded at a rate of $60Hz$ and the prediction interval is four frames, about $66.6ms$, over a duration of approximately $170s$.

The input signals for the x- and z-axis show similar characteristics. Rather long sections feature uniform motion only rarely disrupted by changes in motion. Sections of a constant position are seldom. The y-axis characteristics are slightly different. Large sections with near-constant height and minor head bobbing of less than $5mm$ are interrupted by few stronger vertical movements, e.g. when bending down.

4.1 Specialised Motion Models

The trajectory section used to evaluate the specialised models are chosen from an appropriate linear section of the trajectory (see figure 2 (left)). Table 1 lists the results in comparison to the adaptive Double Exponential Smoothing (aDES) algorithm and no prediction. It can be seen that the aDES significantly reduces the error, but increases jitter. The specialized model, as expected, shows the lowest error and also has a jitter below one, indicating a smoother signal.

For a curved case (figure 2 (right)), the aDES overshoots the apex significantly, and also

linear case	MAE [mm]	median [mm]	0.9-quantile [mm]	jitter
no prediction	12.439	13.084	14.919	1.0
linear model	0.880	0.903	1.302	0.802
aDES	1.843	1.933	3.078	5.747
curved case	MAE [mm]	median [mm]	0.9-quantile [mm]	jitter
no prediction	5.444	4.589	11.506	1.0
exp growth model	0.761	0.547	1.678	0.576
aDES	2.203	1.189	5.422	2.487

Table 1: MAE, median and 0.9-quantile for a linear and curved scenario.

shows high jitter. The specialized exponential change-to-velocity model shows a much lower error and jitter. Notably, the apex is not missed, and predicted positions are smooth.

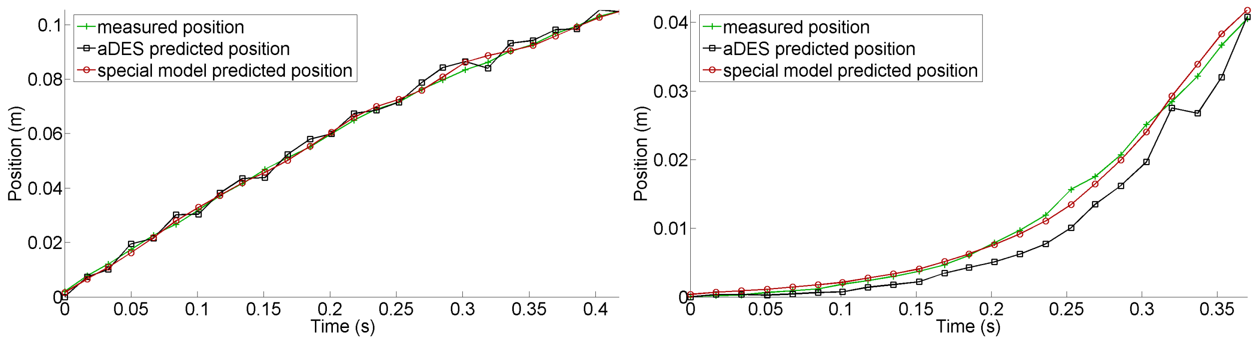

Figure 2: On the left, the prediction results for the linear motion model is shown in comparison to aDES and no prediction. The curved test case used for the exponential growth motion model filter is shown on the right. The prediction interval is $66.6ms$.

These two examples show that specialized models excel their designed motion categories, and outperform general approaches. Of course, this only works for the designed cases - for other movement types, the special models fail. Thus the multiple model approaches combine these special model for the prediction of general movement.

4.2 Adaptive Estimator Results

The 3D position prediction is gained from a combination of separate position predictions per axis. Table 2 shows the error measures over the whole 170 seconds trajectory.

The two multiple model adaptive estimation approaches, MMAE and DMAE, both reduce MAE and jitter compared to the reference aDES implementation. The DMAE approach also shows a definite reduction of the MAE, while the MMAE shows only minor improvement of the MAE. The MMAE often shows more details around apexes, but often assumes curves too quickly. Figure 3 (left) shows two segments from the trajectory and the prediction results from our algorithms. At $0.2s$, the MMAE overestimates the slight increase in velocity and thus spawns another model in response. Because this error can only be recognized some steps later, the predictions from $0.25s$ to 0.3 are off. This can also be noticed in figure 3 (right), where jitter originates from this problem ($0.1s$ to $0.4s$). The DMAE captures fewer small details, but performs globally better.

4.3 Performance Results

Three MMAE instances for motion prediction and one quaternion aDES instance for orientation prediction use on average 0.73ms for one prediction step (AMD FX-8120, MSVC 10). In contrast the DMAEs consumes an average of 0.17ms per prediction. This shows that the gain in accuracy of the DMAE is not obtained at the expense of computation time.

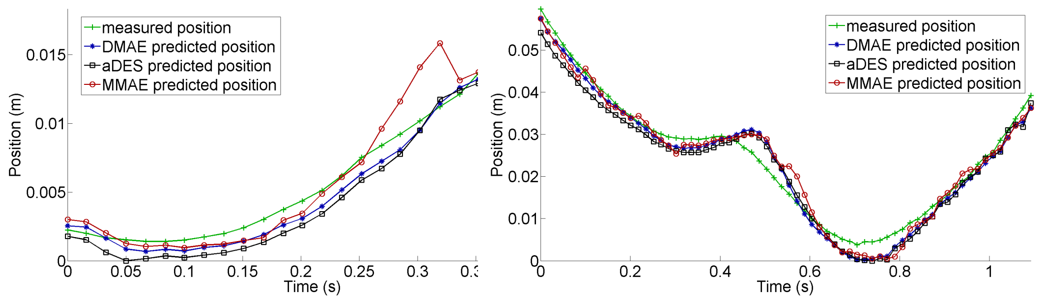

Figure 3: Prediction results for a prediction interval of $66.6ms$. (Left) shows the error-prone reaction of the MMAE to velocity changes. (Right) shows occasional overshooting at curves.

5 Conclusion

In this paper, we have proposed new methods for adaptive prediction of the tracked human head for real-time scenarios. By examining motion patterns, we identified several specialized prediction models that describe different types of human motion. While these specialized models are highly accurate for their designed motion types, they cannot be globally used because the type of motion frequently changes. For this, we proposed two multiple model adaptive estimation approaches that use a combination of models and choose the final output based on the current movement type. One of the models combines many models and variants, and interactively creates and deletes filters, for which we proposed suitable heuristics. The other one uses only two models which proofed to be especially well suited.

We evaluated our results based on recorded movement trajectories and compared them to a reference adaptive Double Exponential Smoothing prediction algorithm. The results show that both models outperformed our reference implementation. Jitter was reduced for both multiple model approaches, and the DMAE also showed lower positional errors. On the one hand, this is caused by the possible ambiguity of trajectories preventing the determination of the correct model. For example, an emerging curved trajectory could be either the start of a continuing curve or the return to a linear movement. Thus, the MMAE has to consider multiple, currently plausible models instead of a single one. Using a weighted average reduces the impact of models not matching the actual future trajectory, but is still influenced by extremal models. On the other hand, the DMAE only chooses between two models. This results in a loss of small detail or delayed reaction to changes, but does not overreact to

	MAE [mm]	median [mm]	0.9-quantile [mm]	jitter
no prediction	14.515	12.735	30.464	1.0
aDES	2.999	2.453	5.917	2.814
MMAE	2.806	2.338	5.255	1.909
DMAE	2.539	2.126	4.822	1.496

Table 2: MAE, median and 0.9-quantile for the prediction process of the 3D position prediction for a prediction interval of $66.6ms$.

potential changes, thus reducing the error. In our opinion, the DMAE provides a better trade-off between detail and accuracy.

For future work, several improvements can be motivated. The filter selection heuristic is a central element for position prediction. When designing it, we focused on movement in the x-z-plane, because most movement takes place there. However, using a specialized MMAE for the y-axis could further increase the accuracy, e.g. by modelling head bobbing. Finally, we only presented methods for position prediction. To predict the orientation of the user, specialized MMAEs should be developed. Another step could be to not only use the data from the tracking system, but also global information like room boundaries or other obstacles. Predicting the orientation quaternion per component using an aDES approach has been tested and delivers promising results. Choosing a MMAE approach should be considered.

In the context of the aforementioned cross-talk cancellation, a low positional error is needed. However, in general the question arises whether users find a high positional error or jitter more problematic, especially for user-centered projections. This would be an interesting topic for future user studies.

References

[ADMR03] Fakhreddine Ababsa, Jean-Yves Didier, Malik Mallem, and David Roussel. Head motion prediction in augmented reality systems using monte carlo particle filters. In *ICAT*, 2003.

[Ass09] Ingo Assenmacher. *Low Latency Technology for Interactive Virtual Environments.* Doctoral thesis, RWTH Aachen University, 2009.

[BH12] Robert G. Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering With MATLAB Exercises.* John Wiley & Sons, 2012.

[CHVN99] Lin Chai, Bill Hoff, Tyrone Vincent, and Khoi Nguyen. An adaptive estimator for registration in augmented reality. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, IWAR '99, pages 23–32, 1999.

[CK13] Min Gyo Chung and Sang-Kyun Kim. Efficient jitter compensation using double exponential smoothing. *Information Sciences*, 227(0):83–89, 2013.

[Gar85] Everette S. Gardner. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1):1–28, 1985.

[HM09] Henry Himberg and Yuichi Motai. Head orientation prediction: Delta quaternions versus quaternions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1382–1392, 2009.

[Jop12] Markus Joppich. *Adaptive Prediction of Human Motion in Virtual Environments using Multiple Model Approaches.* Bachelor thesis, RWTH Aachen University, 2012.

[Kal60] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[KB06] Damien Kelly and Frank Boland. Motion model selection in tracking humans. *IET Irish Signals and Systems Conference (ISSC 2006)*, pages 363–368(5), 2006.

[KM97] David W. Kyger and Peter S. Maybeck. Reducing lag in virtual displays using multiple model adaptive estimation. In *Proceedings of the 1997 American Control Conference*, volume 4, pages 2536–2541, 1997.

[LaV03] Joseph J. LaViola. Double exponential smoothing: an alternative to kalman filter-based predictive tracking. In *Proceedings of the Workshop on Virtual Environments 2003*, EGVE '03, pages 199–206, 2003.

[LSVA07] Tobias Lentz, Dirk Schröder, Michael Vorländer, and Ingo Assenmacher. Virtual Reality system with integrated sound field simulation and reproduction. *EURASIP Journal on Applied Signal Processing*, 2007(1):187–205, 2007.

[Mar06] João C. Martins. Parameters identification with the multiple model adaptive estimation (MMAE) algorithm. In *Proceedings of the 25th IASTED International Conference on Modeling, Indentification, and Control*, pages 501–506, 2006.

[NK13] Thomas Nescher and Andreas Kunz. Using head tracking data for robust short term path prediction of human locomotion. In *Transactions on Computational Science XVIII*, volume 7848, pages 172–191. Springer, 2013.

[SMK98] Kay M. Stanney, Ronald R. Mourant, and Robert S. Kennedy. Human factors issues in virtual environments: A review of the literature. *Presence: Teleoperators and Virtual Environments*, 7(4):327–351, 1998.

[SVL07] Simo Särkkä, Aki Vehtari, and Jouko Lampinen. Cats benchmark time series prediction by kalman smoother with cross-validated noise density. *Neurocomputation*, 70(13-15):2331–2341, August 2007.

[vRM05] Arjen van Rhijn and Jurriaan D. Mulder. An analysis of orientation prediction and filtering methods for VR/AR. In *Proceedings IEEE Virtual Reality 2005*, pages 67–74, 2005.

[Wel09] Gregory F. Welch. History: The use of the kalman filter for human motion tracking in virtual reality. *Presence: Teleoperators and Virtual Environments*, 18(1):72–91, 2009.