2016 IEEE International Symposium on Mixed and Augmented Reality

Learning to Fuse: A Deep Learning Approach to Visual-Inertial Camera Pose Estimation

Jason R. Rambach*
German Research Center for Artificial Intelligence(DFKI)
Augmented Vision Department, Kaiserslautern, Germany.

Aditya Tewari†
German Research Center for Artificial Intelligence(DFKI)
Augmented Vision Department, Kaiserslautern, Germany.
IEE S.A., Contern, Luxembourg.

Alain Pagani‡
German Research Center for Artificial Intelligence(DFKI)
Augmented Vision Department, Kaiserslautern, Germany.

Didier Stricker§
German Research Center for Artificial Intelligence(DFKI)
Augmented Vision Department, Kaiserslautern, Germany.
TU Kaiserslautern, Germany.

ABSTRACT

Camera pose estimation is the cornerstone of Augmented Reality applications. Pose tracking based on camera images exclusively has been shown to be sensitive to motion blur, occlusions, and illumination changes. Thus, a lot of work has been conducted over the last years on visual-inertial pose tracking using acceleration and angular velocity measurements from inertial sensors in order to improve the visual tracking. Most proposed systems use statistical filtering techniques to approach the sensor fusion problem, that require complex system modelling and calibrations in order to perform adequately. In this work we present a novel approach to sensor fusion using a deep learning method to learn the relation between camera poses and inertial sensor measurements. A long short-term memory model (LSTM) is trained to provide an estimate of the current pose based on previous poses and inertial measurements. This estimate is then appropriately combined with the output of a visual tracking system using a linear Kalman Filter to provide a robust final pose estimate. Our experimental results confirm the applicability and tracking performance improvement gained from the proposed sensor fusion system.

Index Terms: I.4.8 [Scene Analysis]: Sensor fusion—tracking; I.2.10 [Vision and Scene Understanding]: Motion—Modeling and recovery of physical attributes; I.2.6 [Artificial Intelligence]: Learning—Connectionism and neural nets

1 INTRODUCTION

Accurate camera pose tracking is a core enabling technology for Augmented Reality (AR) applications using handheld or wearable devices [1]. Precise estimation of the camera's six Degree of Freedom pose (6DoF) consisting of camera position and orientation allows realistic rendering of virtual objects in the observed scene [2].

Vision-based systems for tracking using markers or natural features generally perform well in scenarios with slow camera motion [3, 4, 5]. However, in situations where the image quality is compromised, for example during fast camera movements that cause blurring or during sudden illumination changes, pure visual tracking systems tend to fail. On the other hand, pose tracking using inertial sensors (accelerometers and gyroscopes) is more suitable for following fast motion since the sensors can operate at a much higher frequency, but usually provide biased measurements with high noise levels. For this reason, there has been a lot of research on sensor fusion pose tracking systems attempting to combine measurements from visual trackers and inertial sensors in order to achieve more robust tracking [6, 7, 8, 9, 10, 11].

Commonly, statistical filtering approaches such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) or Particle Filters (PF) are used in sensor fusion systems. A tightly coupled fusion system that processes measurements from the visual and inertial sensors in an EKF framework is proposed in [12]. In their work, four different previously proposed system models for fusion are compared, with some of them using only the gyroscope and others using both inertial sensors. The system model treating the inertial measurements (acceleration and angular velocity) as control inputs to the time update of the EKF is shown to achieve the best performance considering tracking accuracy and computational overhead.

A simultaneous motion and structure estimation system using sensor fusion is given in [8]. Both the EKF and the UKF were used, showing similar tracking accuracy with the EKF being much faster in computation time. A visual tracking marker-based system where inertial tracking is deployed as a substitute only when the visual target is occluded is given in [13]. Another loosely coupled fusion approach is presented in [11] and applied to visual-inertial fusion in smartphones. An adaptive Kalman filter with abrupt error detection is used to fuse the output of an inertial and a visual tracker, however only the case of tracking a planar 2D target is considered. Another apporach is to use the inertial tracking only to provide guidance to the visual tracking system as to where tracked features are expected to be detected [14]. In more recent work, the integration of inertial measurements is done by solving an optimization problem or variations of the EKF [15, 16]. These advanced methods still employ a parametric inertial sensor error model of bias and Gaussian noise.

Adding inertial measurements in a visual tracking framework is a task that requires a lot of preparatory high precision work. A hand-eye calibration consisting of a rotation and a translation between the camera and the inertial sensors has to be computed in order to be able to bring the measurements from the visual and inertial sensors to the same reference coordinate system [17, 18, 19]. This calibration was added to the filtering framework state in [20]. Thus, self-calibration between camera and inertial sensor is performed during operation of the tracking system by adding however additional complexity to the filtering. Another calibration from the inertial measurement unit coordinate system to the global coordinate system has to be computed in order to be able to remove gravity from

*e-mail: Jason_Raphael.Rambach@dfki.de
†e-mail:Aditya.Tewari@dfki.de
‡e-mail:Alain.Pagani@dfki.de
§e-mail:Didier.Stricker@dfki.de

978-1-5090-3641-7/16 $31.00 © 2016 IEEE
DOI 10.1109/ISMAR.2016.19

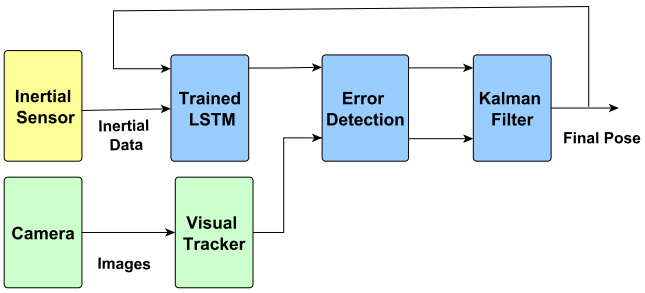

Figure 1: Proposed Fusion System Architecture.

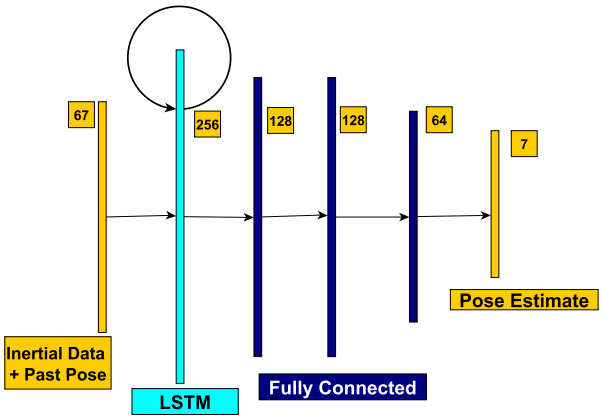

Figure 2: Architecture of the proposed LSTM for camera pose estimation from inertial measurements.

the accelerometer measurements based on the current orientation of the sensor and obtain linear acceleration [12].

Furthermore, in commercially available inertial sensors the measurements are not only disturbed by noise, but have also orientation dependent biases and possibly an incorrect axis alignment [21]. Non-linearity and noise scaling over the measurement range can also be encountered [22]. Modeling this behaviour and estimating noise covariances and biases in order to integrate the measurements in a statistical filtering framework can lead to a very complex system model. Additionally, synchronization of inertial sensors measurements and camera video frames is also desirable.

The main contribution of our work is a novel approach to visual-inertial fusion that uses deep learning techniques in order to model the correspondence of the measurements from an inertial sensor to the tracked camera pose. A long short-term memory (LSTM) [23] is trained to estimate the current camera pose using previous estimates and measurements from the inertial sensors. This allows for rapid deployment of a visual-inertial pose tracking system since a lot of calibration and noise estimation work for the inertial sensors is replaced by a simple LSTM training phase from a set of recorded data. Furthermore, we verify the applicability and tracking accuracy improvement of our sensor fusion tracking in comparison to a purely visual pose estimation tracking system.

To our knowledge, this is the first time that deep learning techniques are applied to the visual-inertial sensor fusion pose tracking problem. In the recent work of Kendall et al. [24] a Convolutional Neural Network (CNN) was trained to solve the problem of 6DoF pose tracking using images as input showing promising results but not achieving the required accuracy for an indoor AR application.

This paper is structured as follows: In the next section we formulate the problem at hand and the made assumptions. In Section 3 our approach to visual-inertial fusion is presented. The architecture of the proposed fusion pose tracking system is explained and its individual components are analyzed. In Section 4, the experimental setup used to evaluate the performance of the proposed system is described and the experimental results are presented and discussed. The paper ends with some concluding remarks.

2 Problem Formulation

This work addresses the problem of 6DoF camera pose tracking using a visual input and corresponding measurements from the inertial sensors. The problem consists of estimating the rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3x3}$ and translation vector $\boldsymbol{t} \in \mathbb{R}^3$ that relates the camera coordinate frame to an object coordinate frame. Using a homogeneous representation of $3D$ points $\boldsymbol{P} \in \mathbb{R}^4$ in the object coordinate system and a homogeneous representation of $2D$ points $\boldsymbol{p} \in \mathbb{R}^3$ in the camera image coordinate system, the camera pose estimation problem is finding a camera pose $[\boldsymbol{R}|\boldsymbol{t}]$ such that the mapping

$$\boldsymbol{p} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{P}, \tag{1}$$

best fits a set of known $3D$ to $2D$ correspondences $C = \{\boldsymbol{P}_i \leftrightarrow \boldsymbol{p}_i\}$ with $\boldsymbol{K} \in \mathbb{R}^{3x3}$ being the camera instrinsics matrix computed from camera calibration. In the case of visual-inertial fusion for pose estimation apart from the set of $3D/2D$ correspondences C, an array $\boldsymbol{S} \in \mathbb{R}^{Nx7}$ of vectors $\boldsymbol{s}_i \in \mathbb{R}^7$ of inertial measurements is available where

$$\boldsymbol{s}_i = [\tau_i, \alpha_{x_i}, \alpha_{y_i}, \alpha_{z_i}, \omega_{x_i}, \omega_{y_i}, \omega_{z_i}], \tag{2}$$

where τ_i denotes the timestamp of an inertial measurement and α and ω denote the acceleration and angular velocity measured in x, y and z direction of the inertial sensor coordinates frame respectively. The dimension N of the inertial measurements array $\boldsymbol{S}$ is defined by the number of inertial measurements that the sensors provide per every camera frame. Thus, the sensor fusion problem consists of finding a function $\mathscr{F}$ that optimally estimates the pose $[\boldsymbol{R}|\boldsymbol{t}]_k$ at camera frame k using the pose $[\boldsymbol{R}|\boldsymbol{t}]_{k-1}$ from camera frame $k-1$, a set C_k of $3D/2D$ correspondences for frame k provided by the visual tracking systems and the inertial sensor measurements $\boldsymbol{S}_k$ collected between frames $k-1$ and k:

$$[\boldsymbol{R}|\boldsymbol{t}]_k = \mathscr{F}([\boldsymbol{R}|\boldsymbol{t}]_{k-1}, C_k, \boldsymbol{S}_k) \tag{3}$$

In the following the quaternion representation of rotation matrices is used, with a quaternion $\boldsymbol{q} \in \mathbb{R}^4$ consisting of four scalar components $\boldsymbol{q} = [q_w, q_x, q_y, q_z]$ and $\boldsymbol{q}(\boldsymbol{R})$ denoting the corresponding quaternion to a rotation matrix $\boldsymbol{R}$. Thus, in the following the pose at camera frame k will be described by a translation vector $\boldsymbol{t}$ and an orientation quaternion $\boldsymbol{q}$ as $[\boldsymbol{t}|\boldsymbol{q}]_k$.

3 Proposed Approach

In this section, the proposed sensor fusion camera pose tracking system using images and inertial data is described. First, the architecture of the entire system is presented in Section 3.1. Subsequently, a description of the individual system components is given in Sections 3.2 to 3.4.

3.1 Fusion System Architecture

The architecture outline of the proposed visual-inertial pose tracking system is given graphically in Figure 1. The data capturing for the system is done by an inertial sensor device comprising an

accelerometer and a gyroscope, and a camera to capture images. We assume that camera and inertial sensors are synchronized in the sense that recording is initiated at the same time for both and the inertial sensors frequency of measurements f_s is a multiple of the camera frame capturing frequency f_c so that $N = f_s/f_c$, $N \in \mathbb{N}$.

When the k_{th} camera frame is captured, the image data of this frame $\mathscr{I}_k$ is passed to the visual tracking module that provides an estimate of the camera pose $[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{vision}}$ consisting of a translation vector $\boldsymbol{t}$ and orientation quaternion $\boldsymbol{q}$, based on detection and matching of visual features from the image. In parallel an array $\boldsymbol{S}_k$ of N vectors of inertial measurements recorded between frames $k-1$ and k as described in Section 2 is passed to the LSTM module. The LSTM also receives as feedback the final pose estimate of the system $[\boldsymbol{t}|\boldsymbol{q}]_{k-1}$ corresponding to the previous frame $k-1$. Based on a number of previous system outputs and inertial measurements the LSTM is trained to provide an estimation of the current camera pose $[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{LSTM}}$.

A comparison module is used to detect failures of the visual tracking system by comparing it to the output of the LSTM. The inertial tracking done by the LSTM can slowly drift away from the correct pose. However, it does not abruptly produce highly erroneous outputs. On the other hand the visual tracking system can output estimates with high error for some frames. In order not to allow these errors to contaminate the fusion system output the comparison module calculates two distance metrics d_t, d_q between the LSTM output and the visual output, where d_t is the euclidean distance between the estimated positions and d_q is the distance between the corresponding angles of the estimated quaternions. If one of d_t,d_q are found to exceed a threshold it is taken as an indication of failure of the visual system, and only the output of the LSTM is passed to the next module. This comparison module is only used as a safeguard against heavily erroneous poses from the visual tracker and is activated very rarely. The system is thus able to retain a correct pose for some video frames even if visual tracking fails. However, because of the drift of inertial tracking, if no correct pose is given from the visual system for a large number of consecutive the system would have to be reinitialized.

Finally, the Kalman Filter module is used to combine the estimated pose from the LSTM with the estimated pose from the visual system and provide a smoothed output. Since both inputs to this module are already given as camera poses, a simple linear Kalman filter model is used in contrast to approaches using the EKF or UKF for pose estimation [12, 8]. The unitary quaternions $\boldsymbol{q}$ representing orientation are always renormalized at the output of the LSTM and the Kalman Filter.

In total, the use of inertial sensors measurements in our system, allows to filter out bad poses of the visual tracking system through comparison and also replace the visual tracking by inertial tracking when the visual tracking fails completely and thus deal with visual target occlusions, motion blur in images, and illumination changes. The trained LSTM model corresponds to a specific hardware type, however the training procedure only has to be applied once if the camera and inertial sensors are rigidly attached to each other. Furthermore, the training procedure can be automated for any given camera-inertial sensor device. An in depth description of the individual components of our fusion system follows.

3.2 Visual Tracker

A visual tracking system using natural features of objects is used in our approach. During registration, ORB [25] features are extracted from given images of $2D$ registration targets, e.g. posters. By setting the positions of these targets in a common three-dimensional coordinate system, the extracted features can be registered as a set of feature descriptors with corresponding $3D$ coordinates in this coordinate system.

During runtime, features are extracted from each received frame image $\mathscr{I}$ and matched to the registered features based on descriptor similarity. Subsequently, a ratio test between first and second match similarity is performed in order to exclude ambiguous matches. Using the selected matched features $2D$ points in the current image and their known $3D$ correspondences from registration, refined pose estimation is achieved within a RANSAC scheme that iteratively solves the perspective PnP Problem while excluding outliers [3].

The visual tracker outputs a camera pose estimate $[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{vision}}$ for every frame consisting of a rotation quaternion $\boldsymbol{q}$ and translation vector $\boldsymbol{t}$ with respect to the registered targets' coordinate system, or an error message indicating failure of the visual tracker. This can be the case when feature matches could not be established due to e.g. image blurring, or when the number of outliers between matches did not allow achieving the convergence criteria of the RANSAC process.

3.3 LSTM

A neural network architecture with an LSTM layer is trained as a regressor. The neural network is such that the input vector includes all the sensor outputs over the time period in which the optical system makes a camera-pose estimation. A ground truth with inertial measurements and accurate camera pose is recorded for training. At time instance k the LSTM receives the estimated camera pose $[\boldsymbol{t}|\boldsymbol{q}]_{k-1}$ system output through the feedback channel as seen in Figure 1 and a number of N measurements $\boldsymbol{S}_k$ from the inertial sensors. N depends on the ratio between the inertial sensor and the camera capturing frequencies, $N = f_s/f_c$. Given this input, the neural network is trained to produce an estimate of the current camera pose $[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{LSTM}}$. The regression estimates of the neural network are made by minimising the error function used in [24]. This is practically a mean square error function with a scaling parameter for the quaternions error in order to bring position and quaternion error values to approximately the same level. Various neural network architectures were trained and the one providing the best results is further described in detail. The training uses the RPROP Algorithm for the optimization process [26].

The architecture of our LSTM implementation is given in Figure 2. The network input is fed into a 67 node fully connected linear layer. The 67 inputs correspond to 10×6 inertial measurements($N = f_s/f_c = 10$ in our implementation) plus 7 inputs for the last output pose of the system (feedback). The input layer is followed by an LSTM layer which captures the temporal relationship between the input. The LSTM layer has 256 nodes. Three more fully connected layers are further added to the network. The output layer is a seven node layer, representing the spatial location $\boldsymbol{t}$ and the orientation quaternion $\boldsymbol{q}$ of the camera. Non-linearity is added to the network by using a *tanh* activation function with each layer apart from the output layer. The LSTM network is unwrapped such that the input is fed as a sequence of vectors generated from the sensors and the camera pose. Decision predictions are made in a moving window style, such that the window slides over the incoming sequence. The network is trained on a training dataset with ground truth camera poses obtained from observation of fiducials and corresponding inertial measurements. The learned function of the LSTM is a subfunction $\overline{\mathscr{F}}$ of the sensor fusion problem function $\mathscr{F}$ defined in Equation 3, without the visual tracking system correspondences set C:

$$[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{LSTM}} = \overline{\mathscr{F}}([\boldsymbol{t}|\boldsymbol{q}]_{k-1} : [\boldsymbol{t}|\boldsymbol{q}]_{k-i}], \boldsymbol{S}_k) \tag{4}$$

$\overline{\mathscr{F}}$ gives a pose estimate in frame k using the previous pose estimates from frames $k-1$ to $k-i$ where i is the depth of the regression and inertial sensor measurements.

3.4 Kalman Filter

Our sensor fusion system uses a linear Kalman Filter to fuse the estimated poses $[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{LSTM}}$ and $[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{vision}}$ from the LSTM and the

visual tracking respectively, into a final system output pose estimate $[\boldsymbol{t}|\boldsymbol{q}]_k$. The filter state is also a vector consisting of a position $\boldsymbol{t}$ and orientation quaternion $\boldsymbol{q}$. During the prediction step of the filter, instead of using the previous state of the filter the pose estimate from the LSTM is used. The prediction equations are

$$[\boldsymbol{t}\hat{|}\boldsymbol{q}]_k = \boldsymbol{F}_k[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{LSTM}} \tag{5a}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \boldsymbol{F}_k\boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{F}_k^T + \boldsymbol{Q}_k, \tag{5b}$$

where $\boldsymbol{F}_k$ is the state transition matrix which we set to an identity matrix, $\boldsymbol{\Sigma}_k$ is the state covariance matrix, and $\boldsymbol{Q}_k$ is the process noise covariance matrix which we set to a diagonal matrix containing the variance of the noise estimated at the outputs of the LSTM. The update of the filter is performed by using the output of the visual tracking system $[\boldsymbol{t}|\boldsymbol{q}]_k^{\text{vision}}$ as the measurement. The equations of the filter update step are

$$\tilde{\boldsymbol{y}} = [\boldsymbol{t}|\boldsymbol{q}]_k^{\text{vision}} - \boldsymbol{H}_k[\boldsymbol{t}\hat{|}\boldsymbol{q}]_k \tag{6a}$$

$$\boldsymbol{E}_k = \boldsymbol{H}_k\boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k^T \tag{6b}$$

$$\boldsymbol{G}_k = \boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k^T\boldsymbol{E}_k^{-1} \tag{6c}$$

$$[\boldsymbol{t}|\boldsymbol{q}]_k = [\boldsymbol{t}\hat{|}\boldsymbol{q}]_k + \boldsymbol{G}_k\tilde{\boldsymbol{y}} \tag{6d}$$

$$\boldsymbol{\Sigma}_{k|k} = (\boldsymbol{I} - \boldsymbol{G}_k\boldsymbol{H}_k)\boldsymbol{\Sigma}_{k|k-1}, \tag{6e}$$

where $\tilde{\boldsymbol{y}}$ denotes the innovation, $\boldsymbol{E}_k$ denotes the innovation covariance, $\boldsymbol{H}_k$ is the observation matrix set to identity here, and $\boldsymbol{G}_k$ denotes the Kalman gain. The state of the filter $[\boldsymbol{t}|\boldsymbol{q}]_k$ after the update is the final pose estimation output of the proposed fusion system for each frame. When the visual tracker fails as explained in Section 3.2, there is no measurement available to perform the update step. In this case, the output of the system is the Kalman filter state after the prediction step $[\boldsymbol{t}\hat{|}\boldsymbol{q}]_k$.

4 Tracker Evaluation

In this section we first describe our experimental setup and sensors used in order to evaluate the proposed sensor fusion system presented in Section 3. In the experimental results section, we first evaluate the LSTM training in terms of error between the estimated pose from the LSTM and the ground truth pose. Subsequently, we evaluate our entire system by presenting a comparison between a purely visual tracking approach using our visual tracker and our sensor fusion approach. This evaluation is done by comparing the overlap between a quadrangle drawn around the 2D target using the estimated pose and a quadrangle drawn using the ground truth pose.

4.1 Experimental Setup

The sensing device used in our experiments is a trivisio colibri inertial sensor (accelerometer and gyroscope) combined with a uEye camera in a common casing as shown in Figure 3. The camera image capturing can be triggered by the inertial sensors using an internal hardware signal in order to capture frames that are synchronized with inertial measurements. Throughout our experiments we set the inertial sensor frequency to $f_s = 100$Hz. The camera is triggered every $N = 10$ inertial measurements and thus captures frames at frequency $f_c = f_s/N = 10$Hz. The resolution of captured images is 1280×1024 pixels. The data received from inertial sensors and camera were directly used in the experiment without any preprocessing. The camera and inertial sensor respective coordinate frames are not aligned and no hand-eye calibration [17] between them was used, as this is indirectly contained in the learned function of the LSTM. The same holds for gravity removal from acceleration measurements. In common sensor fusion approaches that use accelerometer measurements in order to estimate position, the gravity has to be removed from the measurements using the current orientation in order to retain only free acceleration from the measurements [12]. In our approach, this is also learned by the LSTM. A prerequisite for this, is that the direction of gravity in the tracked plane should be the same during the training and the deployment of the system. We aligned the negative z-direction with gravity throughout our experiments. Even if this is not the case, a simple transformation can be applied on the data to ensure this.

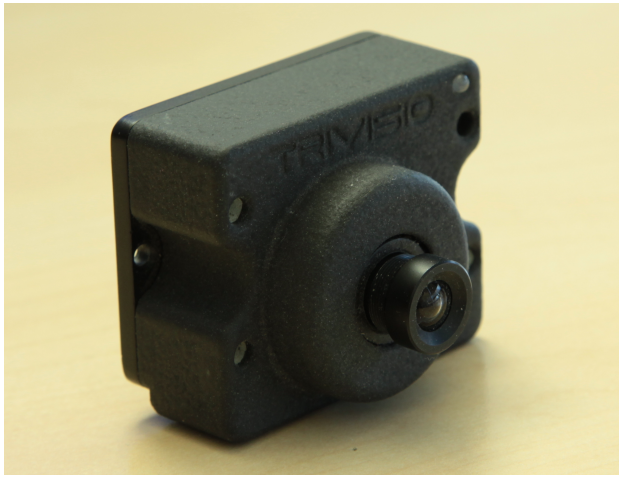

Figure 3: The inertial sensor with camera system used in this work.

Our experimental setup consists of a printed poster image that is surrounded by a number of fiducials as shown in Figure 4. The poster is used as the target of our visual tracking system described in Section 3.2. ORB features are extracted and registered as $3D$ points from the poster and matched in every frame to estimate the camera pose from the $2D/3D$ correspondences. The fiducials are exclusively used for the purpose of obtaining ground truth measurements of the camera pose. These measurements serve for training the LSTM and having a reference for comparison for the evaluation of the system. The proposed tracking system does not require fiducials for pose estimation during runtime. Using three marker fiducials ensures that at least one of them is always visible on the recorded frames to provide ground truth. When more than one fiducials are visible the final ground truth pose is obtained by appropriately combining all fiducials' poses which makes it very robust in our experiments. The used fiducials are described in detail in [27].

During the experiments 3 sets of image sequences and corresponding synchronized inertial measurements were captured using the aforementioned device. Out of these datasets, the first one with a length of 9800 video frames and 98000 inertial sensor measurements was used for training the LSTM, and the other datasets of shorter duration (3389 and 3106 video frames) were used for evaluation of the LSTM and of the system as a whole. The evaluation results are presented next. All datasets were recorded with the capturing device hand-held and contain fast translational motion in all directions, as well as fast rotational motion along all three axes, and combinations of translational and rotational motion.

4.2 Experimental Results

In Table 1 an evaluation of the LSTM training is given. Ground truth poses and inertial measurements are given at the input of the LSTM and the mean absolute error to the ground truth is measured at the output. The error for the sequence used to train the LSTM and two other sequences used was measured as position error (in cm) and orientation error. As expected, the error on the training sequence is lower than the error for the two validation sequences.

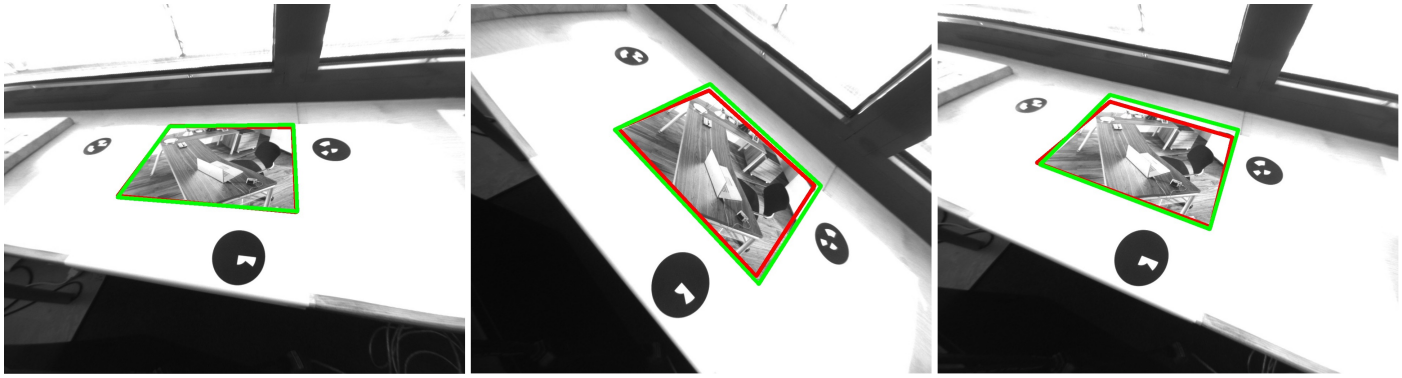

Figure 4: Camera frames during execution of the experiments. The green rectangle is drawn around the expected area of the poster based on the pose obtained from the visual natural feature tracking system. The red rectangle is drawn using the ground truth measurements from the marker fiducials.

Table 1: LSTM pose estimation errors are given as mean absolute errors in position and orientation between the LSTM output and the ground truth pose. The errors indicate that an LSTM can be trained to perform inertial tracking.

	position error			**orientation error**			
	x **(cm)**	y **(cm)**	z **(cm)**	q_w	q_x	q_y	q_z
training sequence	0.708	0.587	0.409	0.0043	0.0021	0.0065	0.0049
validation sequence 1 (slow)	1.695	1.433	0.881	0.011	0.0067	0.0196	0.0135
validation sequence 2 (fast)	1.760	1.492	0.921	0.012	0.0068	0.0206	0.0134

Table 2: Tracking accuracy comparison between a pure visual approach and our proposed visual-inertial tracking system. Overlap corresponds to the average overlap between a quadrangle drawn around the $2D$ tracking target (poster) and a quadrangle drawn based on the ground truth (Figure 4). Failed Frames correpsonds to the number of frames where the system could not provide a pose estimate at all.

	Overlap %	**Failed Frames #**
sequence 1(slow) visual	86.3%	249/3389 7.3%
sequence 1(slow) fusion	90.8%	0/3389 0%
sequence 2(fast) visual	77.1%	487/3106 15.7%
sequence 2(fast) fusion	85.6%	0/3106 0%

However, the errors in general indicate that an LSTM can indeed be succesfully trained to predict camera poses from inertial measurements and previous poses. An increase in the size of the training sequence should naturally lead to a decrease in the presented errors.

The function that has to be learned by the LSTM to model the mapping of inertial measurements to poses is a rather complex one since apart from noise, biases and non-linearities of the inertial sensors, the hand-eye calibration between the camera coordinate frame and the inertial sensor coordinate frame has to be learned as well as the effect of the orientation of the device to the measured acceleration because of gravity. We can also observe that the orientation predictions from the LSTM are more reliable. The reason for this is the higher accuracy of the gyroscope in comparison to the accelerometer and also the fact that the function that has to be learned for the orientation is only dependant on the gyroscope measurements, whereas the function for position estimation needs to take into account the orientation of the device for the removal of gravity from acceleration measurements.

In Table 2 we present an evaluation of the tracking accuracy of our proposed visual-inertial tracking system (Section 3) and compare it to the tracking accuracy of an equivalent purely visual tracking system using the approach presented in Section 3.2. Since the targeted application area of the proposed camera tracking is augmented reality we evaluate the tracking accuracy based on the quality of reprojection on the image by means of the estimated pose using an overlap metric proposed in [11]. For every frame, the four corners of the tracking target ($2D$ poster, see Figure 4) are reprojected to their current image pixel coordinates using the estimated camera pose and a quadrangle is drawn V_{est} to connect them. Similarly, another quadrangle V_{true} is drawn using the ground truth measurements. The percentage of overlap between them is then computed as the intersection of the quadrangles divided by their union given by

$$\text{Overlap}(V_{\text{true}}, V_{\text{est}}) = \frac{|V_{\text{true}} \cap V_{\text{est}}|}{|V_{\text{true}} \cup V_{\text{est}}|} \times 100\%. \quad (7)$$

The average overlap precentage between the visual tracker and the inertial-visual tracker is compared for two sequences of images. One sequence with smoother motion (slow) and one with more abrupt motion (fast). The results show that using the fusion approach an increase can be achieved in the overlap percentage in both the slow and the fast sequence. This increase is mainly due to the fact that inertial tracking provides pose predictions that can be used when visual tracking fails completely in the case when motion blurring or target occlusion occurs. Thus, in the fusion system the missed frames (7.3% in the slow sequence and 15.7% in the fast sequence) were replaced using inertial pose estimations. Apart from the missed frames, there are a few frames where the visual system provides poses with a very high error when there are too many outliers among the matched features. In this case, the comparison of the visual tracker output with the inertial tracking output of the LSTM allows the detection of such false visual system poses and enables the use of the inertial estimation instead. For most of the frames however the visual tracker produces pose estimates that are very close to the ground truth. The Kalman Filter (Section 3.4) process noise levels are set much higher than the measurement

noise levels so that the more precise visual tracking system will be weighted more than the inertial tracking prediction at the filter update step.

Finally, we have measured the average time required by the LSTM to compute a pose prediction to be 3.5 msec. on an Intel Core i7 processor. This verifies that the proposed system can perform pose tracking in real time at very high image frame rates.

5 Conclusion

In this work an alternative approach to visual-inertial camera pose tracking using deep learning techniques is presented. An LSTM is trained to learn the mapping from previous camera pose and ensuing inertial measurements to next camera pose. A linear Kalman filter is then used to fuse the inertial tracking pose estimation to the visual pose estimation and provide a final pose estimate that is robust to fast motion blurring, occlusions, and illumination variations. Most importantly, the proposed system is able to integrate inertial measurements in the tracking framework, without a need for complex modelling of the sensors noises, biases and non-linearities, camera and sensor coordinate frames calibration, and gravity removal from acceleration measurements based on orientation. The temporal data procesing of the LSTM should also allow learning and compensation of timing errors between camera and sensors, however sensor drift cannot be compensated for in the current system implementation. The presented experimental results indicate the succesful training of the LSTM even with a limited amount of data, and the benefits in tracking performance obtained through the integration of inertial tracking in the system. Future work includes training on larger sequences of images and inertial measurements, and integration of the approach in more demanding tracking systems, e.g. a SLAM system. Also, the effects of adding some constraints on the training of the LSTM should be investigated.

Acknowledgements

This work has been partially funded by the Federal Ministry of Education and Research of the Federal Republic of Germany as part of the research project PROWILAN (Grant number KIS3DKI018)

References

[1] W. Barfield, *Fundamentals of wearable computers and augmented reality*. CRC Press, 2015.

[2] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey.," *IEEE transactions on visualization and computer graphics*, 2015.

[3] T. Nöll, A. Pagani, and D. Stricker, "Markerless camera pose estimation-an overview," in *OASIcs-OpenAccess Series in Informatics*, vol. 19, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

[4] S. Hare, A. Saffari, and P. H. Torr, "Efficient online structured output learning for keypoint-based object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1894–1901, IEEE, 2012.

[5] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 590–596, IEEE, 2005.

[6] T. Schön and F. Gustafsson, "Integrated navigation of cameras for augmented reality," in *16th IFAC World Congress, Prague, Czech Republic, July, 2005*, pp. 187–187, 2005.

[7] G. Bleser, C. Wohlleber, M. Becker, and D. Stricker, "Fast and stable tracking for ar fusing video and inertial sensor data," 2006.

[8] P. Gemeiner, P. Einramhof, and M. Vincze, "Simultaneous motion and structure estimation by fusion of inertial and vision data," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 591–605, 2007.

[9] F. Servant, P. Houlier, and E. Marchand, "Improving monocular plane-based slam with inertial measures," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 3810–3815, IEEE, 2010.

[10] G. Bleser, *Towards visual-inertial slam for mobile augmented reality*. Verlag Dr. Hut, 2009.

[11] X. Yang, X. Si, T. Xue, and K. T. T. Cheng, "[poster] fusion of vision and inertial sensing for accurate and efficient pose tracking on smartphones," in *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*, pp. 68–71, Sept 2015.

[12] G. Bleser and D. Stricker, "Advanced tracking through efficient image processing and visual–inertial sensor fusion," *Computers & Graphics*, vol. 33, no. 1, pp. 59–72, 2009.

[13] M. Maidi, F. Ababsa, and M. Mallem, "Vision-inertial tracking system for robust fiducials registration in augmented reality," in *Computational Intelligence for Multimedia Signal and Vision Processing, 2009. CIMSVP'09. IEEE Symposium on*, pp. 83–90, IEEE, 2009.

[14] W. K. Obeidy, H. Arshad, S. A. Chowdhury, B. Parhizkar, and J. Huang, "Increasing the tracking efficiency of mobile augmented reality using a hybrid tracking technique," in *Advances in Visual Informatics*, pp. 447–457, Springer, 2013.

[15] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[16] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual–inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[17] J. Alves, J. Lobo, and J. Dias, "Camera-inertial sensor modelling and alignment for visual navigation," *Machine Intelligence and Robotic Control*, vol. 5, no. 3, pp. 103–112, 2003.

[18] J. D. Hol, *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Linköping University Electronic Press, 2011.

[19] A. Petersen and R. Koch, "Video-based realtime imu-camera calibration for robot navigation," in *SPIE Photonics Europe*, pp. 843706–843706, International Society for Optics and Photonics, 2012.

[20] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 957–964, IEEE, 2012.

[21] I. Aicardi, P. Dabove, A. Lingua, and M. Piras, "Sensors integration for smartphone navigation: performances and future challenges," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 3, p. 9, 2014.

[22] I. Skog and P. Händel, "Calibration of a mems inertial measurement unit," in *XVII IMEKO World Congress*, pp. 1–6, Citeseer, 2006.

[23] S. Hochreiter and J. Schmidhuber, "Lstm can solve hard long time lag problems," *Advances in neural information processing systems*, pp. 473–479, 1997.

[24] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2938–2946, 2015.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.

[26] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *Neural Networks, 1993., IEEE International Conference on*, pp. 586–591, IEEE, 1993.

[27] A. Pagani, J. Koehler, and D. Stricker, "Circular markers for camera pose estimation," in *WIAMIS 2011: 12th International Workshop on Image Analysis for Multimedia Interactive Services, Delft, The Netherlands, April 13-15, 2011*, TU Delft; EWI; MM; PRB, 2011.