Fast and Stable Tracking for AR fusing Video and Inertial Sensor Data

Gabriele Bleser, Cedric Wohlleber, Mario Becker, Didier Stricker
Fraunhofer IGD
Fraunhoferstraße 5
64283 Darmstadt, Germany
{gbleser, cwohlleb, beckerm, stricker}@igd.fhg.de

ABSTRACT

Accurate acquisition of camera position and orientation is crucial for realistic augmentations of camera images. Computer vision based tracking algorithms, using the camera itself as sensor, are known to be very accurate but also time-consuming. The integration of inertial sensor data provides a camera pose update at 100 Hz and therefore stability and robustness against rapid motion and occlusion. Using inertial measurements we obtain a precise real time augmentation with reduced camera sample rate, which makes it usable for mobile AR and See-Through applications.
This paper presents a flexible run-time system, that benefits from sensor fusion using Kalman filtering for pose estimation. The camera as main sensor is aided by an inertial measurement unit (IMU). The system presented here provides an autonomous initialisation as well as a predictive tracking procedure and switches between both after successfull (re)-initialisation and tracking failure respectively. The computer vision part performs 3D model-based tracking of natural features using different approaches for yielding both, high accuracy and robustness. Results on real and synthetic sequences show how inertial measurements improve the tracking.

Keywords: Augmented Reality, Sensor fusion, Tracking.

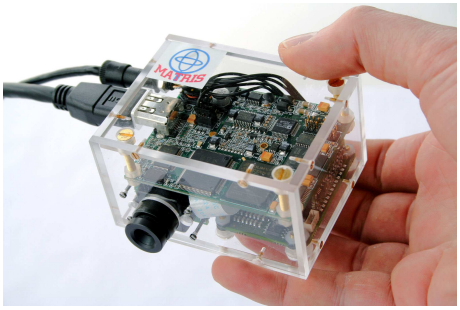

Figure 1: The integrated camera and IMU hardware.

1 INTRODUCTION

To obtain accurate registration of the camera pose, 2D/3D correspondences provided by vision-based tracking algorithms are fused with measurements of a miniature inertial measurement unit (IMU) [7] [1] [16] [6]. The hardware we use is shown in Figure 1. It includes a color firewire camera integrated with a miniature IMU that processes sensor readings at very high update rates and triggers the image grabber, hence supplying a synchronized stream of video and inertial measurements, namely acceleration and angular velocity, which are integrated to obtain relative pose estimates. However, without a constant correction, noise and computational errors lead to a rapid drift in the camera position and orientation. For this reason, it is essential to track and update image features in a reliable way.

Different types of feature detectors [2] [10], descriptors [12] [9], alignment [18] and update methods [8] can be applied. The detection and description of point features can be performed in a robust, Euclidean or even affine invariant manner [11]. This is mostly very time-consuming and not absolutely required during tracking. In some cases it can even be a disadvantage since information gets lost through the abstracted description of the feature. Another approach is to directly use the pixel intensity information, what is less time-consuming but sensitive to illumination and viewpoint changes. However, the latter can be eventually compensated, if predictions of the camera pose are available. A frequent update of the features' descriptors increases their uncertainties and often introduces drift into the camera registration. Hence, the careful selection and long-term tracking of a small set of landmarks is desirable [13].

Rgnqs Bn l l tmhb`shnmr oqnbddchmfr #RAM 7/,75832,/4,3
WSCG'2006, January 30 – February 3, 2006
Plzen, Czech Republic.

2 APPROACH

Our approach differentiates two phases of the vision-based tracking: initialisation and predictive tracking.

The system (re)-initialisation performs the computation of the initial camera parameters within a global reference coordinate system without any assumption about the approximate pose. It has to work robustly without user interventions and has to be very accurate. Errors within the initial orientation are eminently critical, as the acceleration measurements of the IMU include gravity and therefore are interpreted with respect to the orientation (see 3.4). The predictive tracking processes consecutive frames. It has to be very efficient and can take advantage of temporal coherence and the high-quality prediction with the IMU as additional sensor. The pose computation bases on a Kalman filter, which processes the camera data with the measurements of the IMU and generates the best estimate of the current pose.

In [4] we used Euclidean invariant SIFT features [10] for both, initialisation and tracking, and updated the feature map and descriptors constantly. We keep this approach for the initialisation of the tracker only, since very robust results have been reached.

For the on-line tracking we take advantage of the IMU measurements and designed a vision-based tracking module, that applies simple feature tracking. In the first implementation stage, our features (so-called anchors) are 3D patches described by image textures. They are generated offline using structure from motion techniques and build a map of well-defined landmarks, which are highly stable and precisely defined in 3D. We register those anchors within the current image by pre-warping their descriptors with help of the predicted camera pose of the Kalman filter.

3 FRAMEWORK

The run-time system consists of three interacting components and a database, which contains all relevant offline data. The computer vision component is further divided into an initialisation (see 3.2) and a predictive tracking part (see 3.3) and has access to the database for either querying data for initialisation or for tracking. Initialisation yields a 6 DOF pose, whereat the predictive tracking generates 2D/3D correspondences for single frames.

The IMU does not only provide measurements of its acceleration and angular velocity, but is also able to autonomously determine a very precise estimate of its absolute orientation, which is helpful for initialisation.

The sensor fusion component (see 3.4) receives measurements from both sensors, and updates the state of the Kalman filter using the appropriate model with respect to the given kind of data. As the IMU runs at 100 Hz, the state of the Kalman filter is refreshed at the same rate, making a high-quality pose estimation available to the vision-based predictive tracking. Moreover, this component controls the application flow by solely being authorised to switch between initialisation and tracking. If the computer vision component reports failures, the tracking can be holded up with the IMU data over a short period of time before leading to a divergence of the Kalman filter, that leads to reinitialisation. Figure 2 shows the architecture of the system.

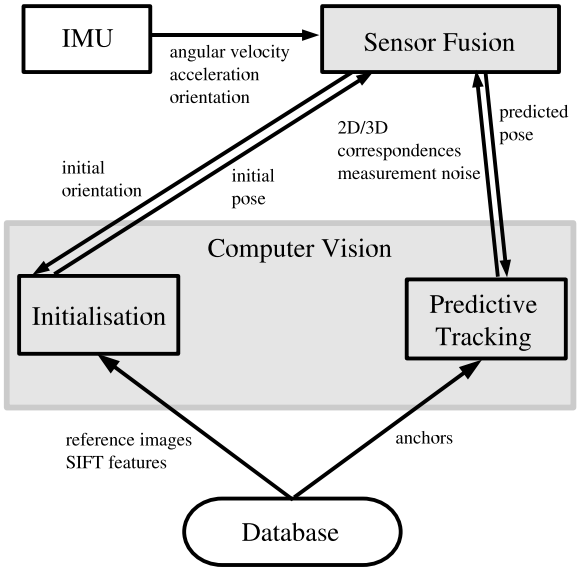

Figure 2: Architecture of the run-time system.

3.1 Offline Preparation

The vision-based component requires some pre-processed input data for initialisation and tracking. This data is generated within an offline step for every 3D scene and camera setup and exported to an xml file for permanent storage and re-use. At system startup, the data is imported to the run-time system by the database component. The database contains:

- Reference images - these consist in fully calibrated images that show the 3D scene from a sufficient number of viewpoints. Furthermore a group of 3D SIFT features is associated to every reference image. They are used during initialisation to get a pose estimate for a given camera frame. Reference images can be either exported by an automatic 3D scene reconstruction step [15] or they can be manually calibrated using a CAD model [4].

- Anchors - they represent persistent features defined in 3D. Currently, we process 3D patches, given as four 3D corners and a surface normal, with an associated image texture that has been sampled from the reference image with the most frontal view onto the plane.

3.2 Initialisation

The initialisation is achieved by matching SIFT features between the live video frame and pre-calibrated reference images, for which the features are known in 3D.

To deal with multiple reference images efficiently, this task is divided into two steps:
Firstly, the database is queried with the current camera frame to find the most likely reference image using a content based similarity measure [14]. At the current implementation, the image retrieval is based on color histograms. Additionally, the initial orientation estimate of the IMU is exploited by finding the closest reference pose with respect to the camera view direction. The roll angle is neglected, as the SIFT features used for obtaining 2D/3D correspondences in the second step, are invariant against in-plane rotations. The second step is carried out with the pre-selected reference image and its associated features as detailed in [4].

3.3 Predictive Tracking

The task of this module is the generation of 2D/3D correspondences by locating the 3D anchors provided by the database within the current camera frame using the pose prediction of the sensor fusion module. This is done within the following steps:

1. query anchors from the database, that are visible from the predicted pose
2. pre-warp their descriptors to the appearance expected from that viewpoint
3. register the anchors within small uncertainty regions around the expected positions using the transformed descriptors
4. establish 2D/3D correspondences by associating each registered 3D anchor centroid to its measured image position
5. perform an outlier rejection to verify valid correspondences
6. pass all valid correspondences to the sensor fusion
7. report a tracking failure, if there have been few inliers
8. update the search radius

As mentioned in paragraph 3.1 we define an anchor as a 3D plane that is described by an image texture. The latter is sampled from a reference image with a preferably frontal view onto that plane. As the reference images are fully calibrated and a pose prediction for the current frame is available, the pre-warping of each anchor description (step 2) is described by a homography

$$H = K(\delta R - \delta T \cdot \frac{\vec{n}^T}{d})K^{-1}$$

where $\delta R, \delta T$ is the relative camera motion between the current prediction and the reference image, the anchor description has been sampled from, and $\vec{n}, d$ are the plane parameters given in the camera coordinate system of the reference image [5]. The camera intrinsics K are assumed to be static. An example anchor description and pre-warping is given in Figure 3 a,b. The closer the prediction comes to the actual pose the more is the current anchor appearance resembled by the pre-warped one and the more confident becomes the registration. But we made the experience that small distorsions do not harm the registration.
Referring to step 3 the pre-warped descriptors are located within the current frame by performing a block matching within a small uncertainty region around the expected anchor positions. The latter are simply computed by projecting the 3D centroids with the current pose prediction. Typically we extract a patch (16x16 or 32x32 pixels) from the warped texture around the centroid and use this for the registration. To be independent of intensity changes like $\alpha f(i,j)$, we use the normalized cross correlation as similarity measure [3]. Computing this measure for all pixel positions around the expected image location within a square search region, we find the position that minimizes the criterium and take this as the correct anchor location for establishing a 2D/3D correspondence. Figure 3 points out this procedure.
As the performance of the block matching (assuming a fixed patch size) depends exclusively on the size of the search region, we keep this as small as possible while assuring a stable tracking. We give a fixed upper T_{max} and lower T_{min} threshold for the search radius R in pixel but choose this parameter dynamically during tracking (step 8). After initialising this parameter with the upper value, we update the search range for the next video image R_{t+1} depending on the maximal observed feature displacement D_{max} using the following rules:

$$\begin{aligned} R_{new} &= D_{max} + T_{min} \\ \alpha &= \frac{D_{max}}{R_t} \\ R_{t+1} &= (1-\alpha)\cdot R_t + \alpha \cdot R_{new} \\ R_{t+1} &= \min(T_{max}, \max(T_{min}, R_{t+1})) \end{aligned}$$

The formulas are interpreted as follows: the weight $\alpha \in [0..1]$ (as $D_{max} \leq R_t$) defines the influence of R_{new}, which merely depends on the measured discplacements (and the constant T_{min}), onto the changeover of R_t, the currently used search radius, to R_{t+1}, the radius for the next frame. If the maximal measured displacement is near to the search radius (D_{max} is near to R_t), the weight onto the meassured displacement gets near to 1 and as a result the search range increases quickly. This is very important as the block matching can't return the correct registration, if it does not lie in focus. If D_{max} is far away from R_t, the weight becomes small, hence the search range decreases slowly.
The set of correspondences resulting from the anchor

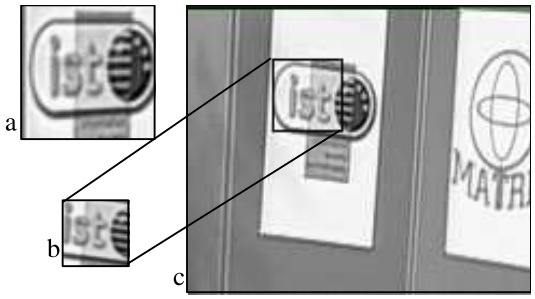

Figure 3: pre-warping and block matching: a: attached image texture as stored in the database, b: pre-warped texture, c: start position for block matching within camera frame

registration are filtered by RANSAC-like outlier rejection [5]. As the RANSAC algorithm also yields a 6 DOF pose, a tracking failure is not only reported, if there have been few inliers, but also if the vision-based pose estimate is completely different from the prediction.

3.4 Sensor Fusion

This component runs a Kalman filter on the measurements from both sensors and makes a pose estimate available to the vision-based tracking. More details can be found in [17] and [7][6].

Camera Motion Model

To fit the Kalman filter framework the motion of the camera is described as a nonlinear system model [17] [7]:

$$x_{k+1} = f(x_k, u_k, v_k)\ ,\ v_k \sim N(0, Q_k) \qquad (1)$$

with state vector $x_t = [p\, v\, q]^T$ representing the camera position, velocity and orientation quaternion in a global coordinate system. This equation is used to predict the pose at each time step by integrating the sensor data $u_t = [\omega\, a]^T$ composed of the angular velocity ω and the acceleration a. As the sensor and the camera have different coordinate systems the data has to be transformed using the so-called hand-eye calibration R_X. We assume that this transformation is only rotational i.e. the camera and sensor center points are very close.

$$\begin{aligned} p_{k+1} &= p_k + v\cdot\Delta t + (R(q)\cdot R_X\cdot a + g)\cdot\frac{\Delta t^2}{2} \\ v_{k+1} &= v_k + (R(q)\cdot R_X\cdot a + g)\cdot\Delta t \\ q_{k+1} &= A(\theta)\cdot q_k \end{aligned}$$

with

$$A(\theta) = cos(\frac{||\theta||}{2})\cdot I_4 + sin\left(\frac{||\theta||/2}{||\theta||}\right)\cdot\Omega$$

$$\begin{aligned} \theta &= \omega\cdot\Delta t \\ \Omega(\omega) &= \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \end{aligned}$$

This integration of sensor data is also known as dead-reckoning.

Every system update increases the pose uncertainty described by a covariance matrix P

$$P_{k+1} = A_k P_k A_k^T + Q_k\ ,\ A_k = \frac{\partial f}{\partial x}(\hat{x}_k, u_k) \qquad (2)$$

by adding the system noise covariance Q resulting from the sensor data uncertainty.

Using Vision Measurements

While the exclusive integration of the sensor data leads to a rapid drift in the pose prediction, the vision-based tracking provides 2D/3D correspondences that are used as measurements to correct the state.

- **2D/3D Correspondences** 2D feature positions are related to the filter state using their 3D points and the camera projection model [17] [7]. The measurement equation for one feature point $\xi_i = (u, v)$ in the camera frame corresponding to the 3D point s_i in the world coordinate system is expressed as

$$\begin{bmatrix} Zu - fX \\ Zv - fy \end{bmatrix} = 0 \qquad (3)$$

  with $[X\ Y\ Z]^T$ the coordinate of the feature point in the camera system i.e. $[X\ Y\ Z]^T = R^T(q)\cdot(s_i - p)$

  The uncertainties of the 2D and 3D feature points have to be modelled as independent normal distribution to fit in the Kalman filter. The covariances of the 2D feature points are given by the block matching.

- **2D/2D Correspondences** 2D/2D correspondences of points or lines could also be used in the Kalman filter by relating the velocity of the features to that of the camera. The derivation of the measurement equation can be found in [17]. The expression for one feature $[\xi_k\ \psi_k]$ is

$$\begin{aligned} [-f\alpha,\ f,\ \xi_k\alpha - \psi_k]\, v_k &= 0\,, \qquad (4) \\ \alpha &= \frac{\dot{\psi}_{Dis} - \dot{\psi}_{Rot}}{\dot{\xi}_{Dis} - \dot{\xi}_{Rot}} \end{aligned}$$

  with $\dot{\xi}_{Dis}, \dot{\psi}_{Dis}$ being the velocity of the feature due to the camera displacement, $\dot{\xi}_{Rot}, \dot{\psi}_{Rot}$ being the velocity due to the rotation and f being the focal length of the camera.

- **Divergence Monitoring** Loss of feature points over a longer period, e.g. because of successive tracking failures, causes the Kalman filter to diverge. As the filter is not able to recover itself, it has to be reinitialized externally (see 3.2). If the variance of the error $h(\hat{x}_k, 0)$ over the last frames excesses the standard gaussian error, this is taken as a hint for an irreparable drift in the filter state and causes the sensor fusion component to request an initial pose estimate from the computer vision part.

Data Synchronization

The data provided by the inertial and vision sensors has to be synchronized to provide a correct real time pose estimation. Two problems have to be solved to obtain proper synchronization:

Multi-Rate sensor fusion The computer vision and the IMU run at different sampling rate. The Kalman filter is able to perform several system updates before new point correspondences are provided. A similar multi-rate sensor fusion problem can be found in the vehicle navigation literature where GPS position updates are not available as often as inertial data. In our system, the camera shutter is triggered by the IMU so that the camera images are synchronized with the inertial data at hardware level. As the IMU is running at 100 Hz, the camera can be run at 25 or 12.5 Hz.

Lag of vision data The computer vision (CV) takes some time to process a camera frame (Figure 4). So the point correspondences are delayed when they are provided to the sensor fusion. The Kalman filter performs system updates (SU) for each IMU data sample. As long as no vision data is received the system state and IMU data are buffered. When point correspondences are available, the system state is reverted to the vision data timestamp and updated to current time with the buffered IMU data.

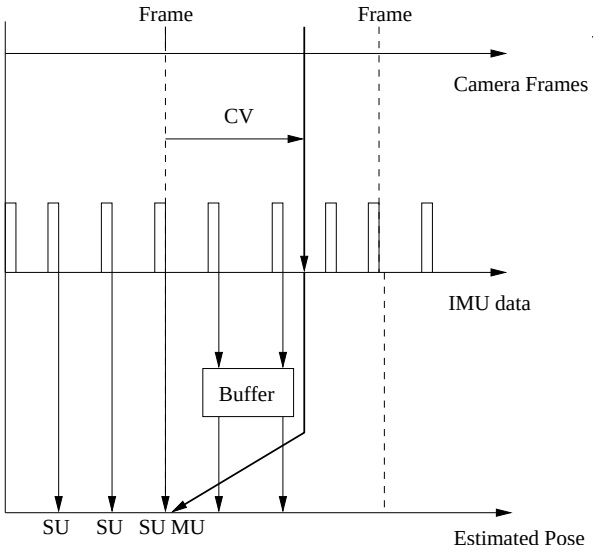

Figure 4: Synchronization of the run-time system.

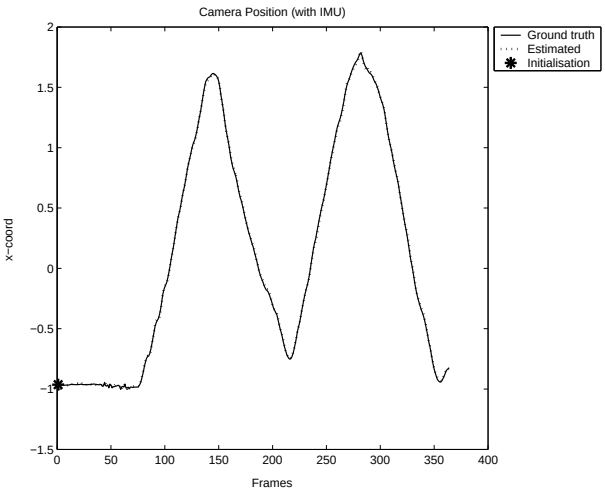

Figure 5: Position (x coordinate) of the camera with use of IMU data, vision data at 5 Hz.

4 RESULTS

We tested our system on live video using the hardware represented in section 1. Figures 9-10 show some augmentations from the live video, where the camera has been triggered with 12.5 Hz. The system has also been tested on synthetic data to provide repeatable results. Some synthetic offline sequences have been generated with resolution 720x576 pixels. The observed 3D scene is a textured 3D model of a room. Four reference images and 40 planar texture patches (anchors) have been pre-processed as input for the vision-based initialisation and the predictive tracking. As the scene has been rendered from a 50 Hz camera track, the ground truth pose data is available at 50 Hz and can be compared to the estimated pose of our system. The inertial sensor data has also been computed from the virtual camera track and simulates the synchronized 100 Hz samples of an IMU moving on this track.

Data fusion at low vision rates

The system has been tested at different vision data rates from 5 to 50 Hz. The improvement of the tracking using the IMU data is most visible at low vision rates. Figures 5 and 6 show the estimated camera position in comparison with the ground truth data with and without the use of IMU data.

Without the IMU support, the pose computation is less accurate and the Kalman filter even diverges so that several re-initialisations are needed. The IMU data allow to track rapid camera movements which are difficult to follow with a purely vision based system.

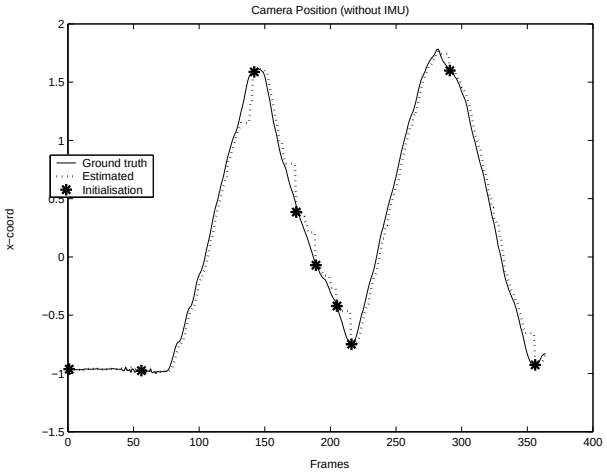

Figure 6: Position (x coordinate) of the camera without use of IMU data, vision data at 5 Hz.

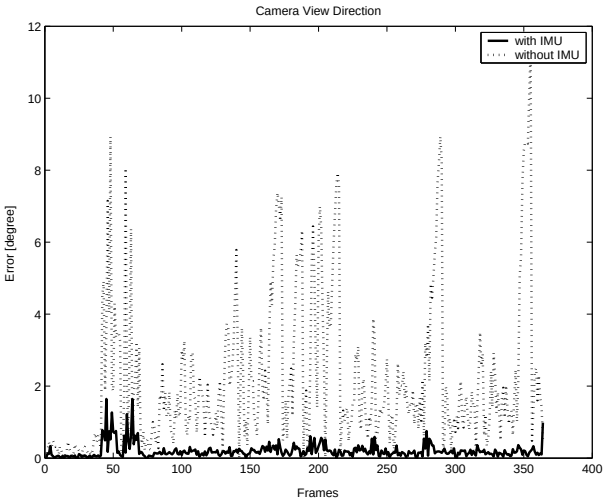

Figure 7: Orientation error of the camera with and without use of IMU data.

Figure 7 plots the orientation error with and without the use of the IMU data. As the IMU gives very accurate gyroscopic measurements, the system is able to compute the rotational part of the camera pose very precisely.

Pose prediction

As the IMU delivers an update rate of 100 Hz, the camera pose is updated at the same rate. Hence the current pose is always available to the system so that it can provide a high-quality prediction of the feature positions to the vision part. The search ranges of the block matching can then be reduced, as shown in figure 8.

5 CONCLUSION

We designed a flexible run-time system for real-time 3D camera tracking fusing vision-based data with

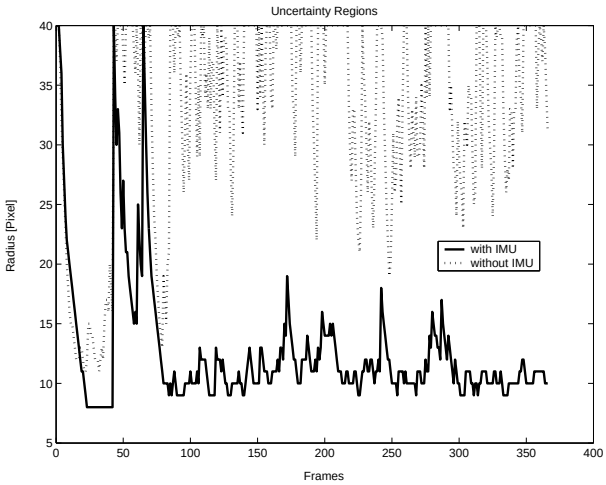

Figure 8: Search ranges needed by the block matching.

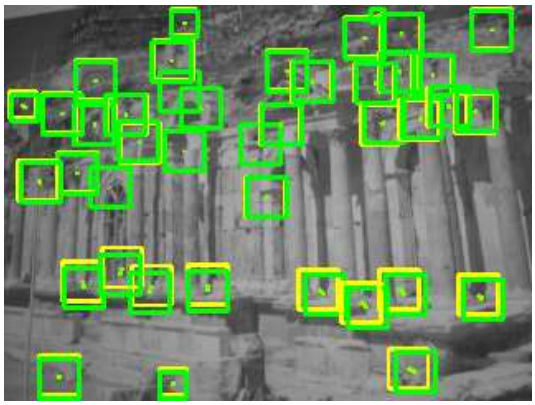

Figure 9: Predictive Tracking: predicted and matched patches are overlayed.

Figure 10: Video frame with live augmentation. The texture is correctly overlayed on the video image using the computed camera pose from the system.

inertial sensor data. As the IMU provides angular velocity and acceleration measurements at a refresh rate of 100 Hz, the camera pose is updated at the same rate. Simultaneously, the camera sample rate can be reduced, whereat a high-quality pose prediction is nevertheless passed to the vision-based tracking.
We showed, that the integration of inertial data improves the tracking significantly, while minimizing CPU costs and yielding high precision and rendering frame-rate, which is crucial for mobile See-Through applications. At the current state the vision-based tracking module operates exclusively with 3D planar features that have been generated offline. Therefore the tracking is restricted to parts of the environment that have been reconstructed within the preparation step. Future work includes the online reconstruction of temporal features, which are less confident but hold up the tracking, whenever the camera observes additional territory.

6 ACKNOWLEDGEMENTS

This work has been performed within the MATRIS consortium, a research program within the European Union (IST-002013). The authors would like to thank the EU for the financial support and the partners within the consortium for a fruitful collaboration. The synthetic image sequences have been provided by the university of Kiel, the synthetic sensor measurements have been provided by Xsens Motion Technologies. For more information, please visit its website, `www.ist-matris.org`.

REFERENCES

[1] L. Armesto, S. Chroust, M. Vincze, and J. Tornero. Multi-rate fusion with vision and inertial sensors. *Robotics and Automation, 2004. Proceedings. ICRA '04*, pages 193– 199 Vol.1, 2004.

[2] A. Baumberg. Reliable feature matching across widely separated views. *In Computer Vision and Pattern Recognition, Volume 1*, pages 774–781 vol.1, 2000.

[3] O.D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[4] G.Bleser, Y.Pastarmov, and D.Stricker. Real-time 3d camera tracking for industrial augmented reality applications. *In Proc. of WSCG 2005*, pages 47–54, 2005.

[5] R. Hartley and A. Zissermann. Multiple view geometry in computer vision. 2000.

[6] J. Hol. Sensor fusion for camera pose estimation. *Master Thesis, University of Twente*, 2005.

[7] J. Hol, P. Slycke, T. Schoen, and F. Gustafsson. 2d-3d model correspondence for camera pose estimation using sensor fusion. *In Proc. of InerVis workshop at the IEEE International Conference on Robotics and Automation*, 2005. `http://www.xsens.com/`.

[8] I. Takahiro I. Matthews and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 810–815, 2004.

[9] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. *In Proc. Computer Vision and Pattern Recognition*, pages 506–513, 2004.

[10] D. G. Lowe. Object recognition from local scale invariant features. *In Proc. of the International Conference on computer Vision ICCV*, pages 1150–1157, 1999.

[11] K. Mikolajczyk and C. Schmid. An affine invariant point detector. *In European Conference on Computer Vision, Volume 1*, pages 128–142, 2002.

[12] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *In Proc. Computer Vision and Pattern Recognition*, pages 257–264, 2003.

[13] N. Molton, A. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. *In Proc. of British Machine Vision Conference*, 2004.

[14] S. Philippi and G. Bleser. A framework for the comparison of similarity measures in multimedia databases. *Accepted for publication in Proc. of the International Conference on Imaging Science, Systems, and Technology*, 2005.

[15] R.Koch, J.-F. Evers-Senne, J.-M. Frahm, and K.Koeser. 3d reconstruction and rendering from image sequences. *In Proc. of WIAMIS*, 2005.

[16] U. Neumann S. You and R.T. Azuma. Hybrid inertial and vision tracking for augmented reality registration. pages 260–267, 1999.

[17] T. Schoen and F. Gustafsson. Integrated navigation of cameras for augmented reality. *International Federation of Automatic Control (IFAC) World Congress, Prague*, 2005.

[18] T. Zinßer, C. Graeßl, and H. Niemann. Efficient feature tracking for long video sequences. *In Proc. DAGM Symposium*, pages 326–333, 2004.