Augmented Reality in Forest Machine Cabin

Tuomo Palonen*, Heikki Hyyti*†, Arto Visala*

** Autonomous Systems Research Group, Department of Electrical Engineering and Automation, Aalto University, P.O. Box 15500, 00076 Aalto, Finland (e-mail: tuomo.palonen@aalto.fi, heikki.hyyti@aalto.fi, arto.visala@aalto.fi)*
† Department of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute, National Land Survey of Finland, Geodeetinrinne 2, 02430 Masala, Finland (Tel:+358-29-530-1100; e-mail: heikki.hyyti@nls.fi)

Abstract: Augmented reality human machine interface is demonstrated in the cabin of a forest machine outdoors for the first time in real time. In this work, we propose a system setup and a real-time capable algorithm to augment the operator's visual field with measurements from the forest machine and its environment. In the demonstration, an instrumented forestry crane and a lidar are used to model the pose of the crane and its surroundings. In our approach, a camera and an inertial measurement unit are used to estimate the pose of the operator's head in difficult lighting conditions with the help of planar markers placed on the cabin structures. Using the estimate, a point cloud and a crane model are superimposed on the video feed to form an augmented reality view. Our system is tested to work outdoors using a forest machine research platform in real time with encouraging initial results.

Keywords: Attitude algorithms, Augmented reality, Calibration, Forestry, Position estimation, Real time, Sensor fusion

1. INTRODUCTION

Forestry is an important field of industry, especially in Finland. Of all the exports in Finland in 2014, forest products comprised 20.1% and were worth of €11.2 billion (Finnish Customs, 2015). In the same year, the total worth of forest products exports in the world was €338 billion (FAO, 2016). Forestry is multidisciplinary work. It contains not only logging and processing of trees, but also active control of the growth and health of forests.

Forest machines are getting more autonomous. Today, modern Cut-To-Length (CLT) forest harvesters are able to fell, delimb (remove branches) and cut trees to logs on site semi-autonomously. The logs are then collected and delivered to roadside by transporting vehicles called forwarders. Automating processes has proved to increase productivity and quality of the work. However, the challenging environment and complexity of the work slows this development. According to Billingsley et al. (2008) we are not expecting to see fully autonomous forest harvesters in the near future.

Thus, it is important that the forest machine operator is provided information about what the machine has learnt and planned. The problem is that the digital data we would like to show to the operator, such as models of the trees, comes from a dynamic three-dimensional environment. Therefore, conventional sensory information on flat screens is not ideal. The aim should be that the digital data is shown as a part of the real environment allowing a fast and natural way of interacting with the machine, and that is exactly what augmented reality (AR) does.

AR integrates digital data into real world in real time. The digital information can be merged into a live video or it can be augmented to user's perception of the environment. The well-known definition by Azuma (1997) states that every AR system has the following characteristics:

- Combining real and virtual
- Being interactive in real time
- Registering real and virtual objects in 3D

There is evidence that AR can improve the performance of the user in traditional industrial tasks like assembly work (Tang et al., 2003) and maintenance (Henderson and Feiner, 2009). We believe that the technology could also improve productivity in the field of forestry.

In this work, a prototype AR application is implemented for a forest machine research platform (Kalmari et al., 2013b; Kalmari et al., 2014), and it is tested outdoors as shown in Fig.1. The initial system augments point cloud data from the environment and a wire-frame model of the forestry crane into a live video feed displayed on a PC. A machine vision camera is attached to a side of the helmet the operator is wearing in order to get similar movements as using a head-mounted display.

This is to our knowledge, the first time augmented reality is demonstrated in the cabin of a forest machine in real time. Previously, the use of head-mounted displays and augmented reality for forest machine operators has been studied in the work by Nordlie and Till (2015). However, their experiments were carried out in a simulator environment without head tracking.

Fig. 1. Forest machine research platform and the system setup used in the work. A dummy tool was used in the test to play around obstacles added to the scene.

The focus of this work concerns about measuring and estimating accurately the orientation and location of the user's head inside the cabin while the operator is working normally. Accurate pose estimation is important in order to project 3D data from the environment to the respective place on the 2D image on the display. According to Zhou et al. (2008), the combination of vision and inertial sensors has been a great interest in AR pose estimation because they have complementary characteristics to each other. It is also the main reason why these sensors are selected to be used in this work.

For pose estimation, we are using methods from the ArUco library (Garrido-Jurado et al., 2014) with 2D barcode markers, detected by the camera, distributed inside the cabin. There are also similar AR libraries available like ArToolkit (Kato, H. and Billingurst, M., 1999) and ALVAR (VTT, 2009). However, ArUco is selected for this work because it is implemented on top of OpenCV that provides a large inventory of machine vision algorithms.

In this work, the orientation estimate is further improved by combining the vision estimate with inertial measurements. A Kalman filter approach is utilized in the sensor fusion process. In our experiments, we evaluate the performance of the prototype by comparing the augmentation success visually to measurements of the pose.

2. METHODS

2.1 System Setup

The system (see Fig. 1) consists of both pose estimation relative material and hardware for generation of virtual content. There are seven 2D ArUco barcode markers (Garrido-Jurado et al., 2014) attached inside the tractor cabin and they are used for the head localization process. Furthermore, the selected machine vision camera (DFK 41AU02.AS from The Imaging Source) gives 15 frames per second in the Bayer8 format. The lens on the camera (Tamron 13FM22IR) has a 118.6° horizontal angle of view and 90.0° vertical angle of view.

The chosen MEMS inertial measurement unit (IMU) is a MPU-6050 (InvenSense, 2010) with a three-axis accelerometer and a three-axis gyroscope. It communicates with a microcontroller via I^2C-interface at 400 kHz. The measurement range for accelerometer is set to be ± 8g (1g ≈ 9.81 m/s^2) and for gyroscopes ± 500 deg/s. The microcontroller is Teensy 3.1 (PJRC, 2013) which is connected to the IMU on a self-built printed circuit board (PCB). We are using an I2C Device Library (Rowberg, 2016) to collect the inertial measurements with the microcontroller and send them to PC via USB-connection. Both the camera and the PCB are attached rigidly together.

On the other hand, the forest machine research platform consists of a Valtra T132 agricultural tractor and a Kesla 305T forestry crane (Kalmari et al., 2013b; Kalmari et al., 2014). It is ISO 11783 compliant platform and it enables joysticks to control the crane. The four degrees of freedom of the crane are all instrumented with sensors to obtain measurements from each of them. There is also a freely hanging dummy tool used for testing purposes. Its position is measured by two IMUs which are located on the tip of the boom and on the tool itself (Kalmari et al., 2013a).

Additionally, a lidar (2D laser scanner) is attached on the top of the basis of the crane vertically so that it scans the environment from ground to sky (see Fig. 1). The lidar is a LMS221 (SICK, 2008) with a 180 degree scanning angle. As the crane moves, the lidar gets new information which is used to constantly update a 3D point cloud. It is later augmented into the operator's view.

2.2 Calibration

It is important that all the sensors are calibrated to give trustworthy measurements, but in this work we only consider those regarding pose estimation. The camera is calibrated using a Camera Calibration Toolbox for Matlab (Bouguet, 2004). The camera calibration parameters are calculated from 15 planar checkerboard images which are taken from different views. The images are displayed on a television screen in its native resolution to get accurate measures of the pattern. An equidistance camera model is used instead of the traditional pinhole model due to the fish-eye lens on the camera. The accuracy of the calibration result is estimated by root mean square error (RMSE) of the reprojected corner points. RMSE of ~ 1 pixel was estimated.

For calibration of the MEMS IMU, a temperature based calibration method by Hyyti and Visala (2015) is utilized. The measurement model is formulated as

$$f_{meas,i} = p_{gain}^{f_i}(T) f_i + p_{bias}^{f_i}(T)$$
$$\omega_{meas,i} = p_{gain}^{\omega_i}(T) \omega_i + p_{bias}^{\omega_i}(T) \quad (1)$$

$$p_{gain/bias}^{f_i/\omega_i}(T) = a_{gain/bias}^{f_i/\omega_i} T + b_{gain/bias}^{f_i/\omega_i}, \quad (2)$$

where $f_{meas,i}$ and $\omega_{meas,i}$ are the measured accelerometer and gyroscope readings in i:th axis $i \in \{x,y,z\}$ respectively. The raw measurements f_i and ω_i are corrected by the estimated gain/bias values given by the function $p_{gain/bias}^{f_i/\omega_i}(T)$ in (2) which is dependent on temperature *T*.

Practically, the IMU is first cooled down and then let to gradually heat in six different stationary orientations. Then a linear model is fit to each of the measurement sets using the linear least squares method. For the accelerometer, non-temperature dependent gain and bias values are estimated in two different temperatures using an algorithm by Won and Golnaraghi (2010). Then a parametrized line is fitted into those two points in order to obtain the calibration parameters $a_{gain/bias}^{f_i/\omega_i}$ and $b_{gain/bias}^{f_i/\omega_i}$. Only the bias value is estimated for the gyroscope in this work, because the gyroscope measurements are corrected online in the attitude estimation algorithm described later. Because of the missing gyroscope gain calibration measurement, the gain is assumed a constant 1.

It is also essential to localize the 2D markers inside the cabin in the same coordinate system. Probably the best solution would be to precisely mount the markers on to some predefined and measured places. However, this would require a cabin especially designed for this type of a system. Now, the markers are just distributed evenly on to the structures of the cabin in such a way that they do not disturb the operator's work.

The idea is to calibrate the locations of the markers using a vision based method since a camera is already a part of the system. Our method mainly follows the paper by Siltanen et al. (2007). From the four corner points of a rectangular marker, we can calculate a pose between the marker and the camera. Then, if two markers are detected in the same image, we can use the properties of the transformation matrices to calculate a pose between the markers:

$$ {}^{M_1}\mathbf{T}_{M_2} = {}^{C}\mathbf{T}_{M_1}^{-1}\,{}^{C}\mathbf{T}_{M_2}. \tag{3} $$

In (3), $\mathbf{T}$ is a transformation matrix, *C* is a camera coordinate system, and M_n is a marker frame of the n:th (n = 1,2,3…) marker. Since we need the markers to be described in the same coordinate system, we choose one corner point of a preselected marker (main marker) to be the origin of the reference frame. All the other marker locations are projected through a transformation chain on to the chosen frame. Because the errors will accumulate at each step of the transformation chain, the shortest path is chosen by Dijkstra's algorithm (Dijkstra, 1959).

During the automatic marker localization process, the poses between detected marker pairs are constantly updated when new information comes available. Averaging the translation part is trivial. However, since rotation matrices are characterized as orthogonal matrices with a determinant of one, the transformation matrices cannot be averaged elementwise. Thus, the translation and rotation parts of the transformation matrix are split, and the rotation matrices are transformed into quaternions. Markley et al. (2007) proposed an exact solution for averaging quaternions based on minimizing a weighted sum of squared Frobenius norms. However, as the differences between rotations in our case are small, we use an approximate solution also provided by Markley et al. (2007), where quaternions are averaged using

$$ \bar{\mathbf{q}} \equiv \left(\sum_{i=1}^{n} w_i\right)^{-1} \sum_{i=1}^{n} w_i \mathbf{q}_i, \tag{4} $$

where $\bar{\mathbf{q}}$ represents an averaged quaternion and w_i is the weight of the i:th $\mathbf{q}_i$ quaternion.

The selected method has two known flaws (Markley et al., 2007). The first flaw is that the resulting quaternion is no longer a unit quaternion. There exists an easy ad hoc method to fix it, where the averaged quaternion is divided by its own norm. The second flaw comes from the fact that quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. The algorithm to calculate mean is implemented recursively and thus the second flaw can be solved by taking a dot product of the two quaternions under processing. If the dot product gives a negative value, then the newest quaternion is inversed. The weights for quaternions are obtained from the ArUco's sub-pixel corner detection algorithm.

2.3 Head pose estimate

As already mentioned, we are using a combination of vision based pose estimation with a single camera and attitude estimation with a MEMS IMU. The camera pose estimation method is based on the ArUco module in OpenCV with some modifications. The marker detection process follows the procedure presented by Garrido-Jurado et al. (2014). The region size for adaptive thresholding of the grey image has to be chosen carefully in the outdoor environment to keep the markers detectable. We also use a sub-pixel corner refinement option provided by the ArUco module from which a quality value for the detected corners can be estimated.

When all the corner positions are known in the same reference coordinate system, a pose can be calculated from any marker or combination of markers that are detected. ArUco first estimates an initial pose value which is further refined by an iterative algorithm. From a single marker, we get four projective 2D to 2D point correspondences, from which a 3x3 homography matrix can be calculated. The pose can then be extracted from the homography matrix because the intrinsic parameters of the camera are known. If more than one marker is detected and the markers do not belong to the same plane, a direct linear transformation is used to calculate an initial pose estimate. The initial estimate is refined by using a Gauss-Newton iterative algorithm to minimize the reprojection error of the corner points.

Some practical modifications and additions were done to the code provided by ArUco. The algorithm expected a pinhole model, but an equidistance model was used in this work due to the fisheye-lens. Fortunately, OpenCV provides compatible algorithms and for that reason the change was not laborious. It was also noticed that the original iterative algorithm for pose estimation did not change the output of the initial closed form solution. Therefore, it was reimplemented

to the Gauss-Newton algorithm. We also modified the code to allow different sizes of markers to be used together, because the distances of the markers from the operator differ inside the forest machine cabin. The final practical feature implemented, is to discard markers with a quality value below a certain threshold. This feature is important as the pose estimate is highly affected by the accuracy of the detected corners on the image. However, the selection of the threshold value is only based on experience.

A MEMS IMU is also utilized in the pose estimation as it has some advantages over vision based sensors. It can give accurate measurements even in fast motions, it is immune to occlusions, and it has a high measurement rate. However, low-cost MEMS IMUs are affected by significant amount of noise. Other considerable challenges are bias and gain errors that drift over time, and are affected by temperature changes. Hence, an attitude estimation algorithm that fuses accelerometer and gyroscope measurements is needed. The algorithm should also correct gyroscope bias online. According to Hyyti and Visala (2015), there are only a few accurate algorithms that can estimate gyroscope bias online with only triaxial gyroscope and accelerometer. As their implementation is the only one freely available, it is selected for this work.

Hyyti and Visala (2015) use a direct cosine matrix (DCM) based adaptive extended Kalman filter (EKF) for attitude estimation with low-cost MEMS IMUs. The attitude is estimated only partially as only the last row vector of the DCM (corresponding to the gravity vector) is used as one of the states of the sensor fusion algorithm. Neither the gyroscope nor the accelerometer can give any information about the absolute heading of the device. Hence, the heading angle is integrated directly from the bias corrected gyroscope measurements. We will show how to adjust the heading angle later when the vision based estimate is sensor fused with the IMU attitude estimate. One of the main ideas of the selected algorithm is to use a varying measurement covariance $\mathbf{R}_k$ of acceleration measurements. This minimises errors caused by transient non-gravitational accelerations (head movement) in the attitude estimate. The measurement covariance is formulated in (Hyyti and Visala, 2015) as follows:

$$\mathbf{R}_k = (\|\mathbf{a}_k\|\sigma_a^2 + \sigma_f^2)\mathbf{I}_3, \tag{5}$$

where $\|\mathbf{a}_k\|$ describes the magnitude of the estimated non-gravitational acceleration. This estimate is the difference between the predicted direction of gravity and the measurement of the accelerometer. The parameter σ_f^2 represents the variance of the acceleration measurement noise and the parameter σ_a^2 is on the other hand used as a scaling factor for $\|\mathbf{a}_k\|$.

2.4 Sensor fusion of attitudes

Before we can combine the attitude estimates from the camera and IMU, the measurements from the devices must be synchronized. As the selected camera does not provide an external triggering option, the synchronization has to be done on the software level. IMU measurements with the corresponding time stamps and DCM estimates are buffered into a container. When an image is taken, the timewise closest DCM estimate is selected for sensor fusion. However, the PC time stamps, which are taken instantly when an image or IMU measurement is available, do not correspond to real time. So, the problem is to find that time delay.

Our solution is to measure the positions of the IMU and camera in a testing rig, where the IMU and camera are moved horizontally along one of the IMU's axes. The position of the IMU is obtained by double-integration. On the other hand, a ruler is put in front of the testing rig from which the position of the camera can be measured. The small errors of the accelerometer measurements will accumulate when they are double-integrated. Thus, the absolute distance that is calculated for the IMU is slightly different than what it is for the camera. This is solved by normalizing the travelled distance from 0 to 1 for both of the sensors. Finally, third-order polynomials are fit to the discrete measurements, and a time delay that minimizes the squared error differences between the positions is found.

Another request for sensor fusion of the IMU and camera is that their DCM (or rotation matrix) estimates should be expressed in the same coordinate system. As the IMU's measurements are relative to the Earth frame and the camera's measurements are relative to the main marker, we need to know the pose of the main marker in the Earth frame. In this work, this is partly omitted due to lack of measurement data. The main marker is just attached to the tractor cabin, manually aligned with the direction of gravity. However, the movement of the cabin affects this setup, which will show up in the results (see Fig. 5). One easy solution would be to measure the motions of the cabin with an additional IMU.

The sensor fusion module assumes that the attitude estimates of the IMU and camera are synchronized and expressed in the same coordinate system. Because the IMU algorithm estimates the last row of DCM, we use a second filter in cascade to estimate also the first row of DCM. A similar idea is used in the work by Phuong et al. (2009), but we also include the last row estimation in the second filter since the camera can provide additional information. Because the IMU and camera have different sensor rates, the prior estimate comes from the faster IMU. The system has six states which are the first and last row elements of the DCM estimate. A simple state transition is used

$$\begin{aligned} \mathbf{x}_k &= \mathbf{x}_{k-1} + \mathbf{w}_k \\ \mathbf{w}_k &\sim N(0, \mathbf{Q}_k) \end{aligned}, \tag{6}$$

where $\mathbf{x}_k$ is a state vector and $\mathbf{w}_k$ represents a zero mean Gaussian white noise with a process covariance matrix $\mathbf{Q}_k$. It is assumed that $\mathbf{Q}_k$ has no cross-correlation between the states. The variances of state predictions are approximated by measuring how much the estimates of the IMU DCM algorithm vary, when the device is held stationary. The measurement covariance matrix $\mathbf{R}_k$ is obtained similarly. The

values are evaluated from the variances of the estimates provided by the camera pose algorithm. However, the variances are smaller when more markers are visible in an image. It is experimentally noticed that the estimated variances are not significantly reduced after more than three markers are detected. So, the values are calculated for situations where one, two, and three or more markers are visible. Initially, there is no knowledge about the absolute heading. Thus, the state vector elements corresponding to the first row of DCM should have high uncertainty values.

In order to maintain the properties of a rotation matrix, a renormalization method (Premerlani and Bizard, 2009) is utilized. The first and last row vectors **X** and **Z** of DCM should be perpendicular, and thus have a zero dot product. The value from the dot product is the amount of error and it is reduced by cross-coupling:

$$\mathbf{X} \cdot \mathbf{Z} = error, \tag{7}$$

$$\begin{aligned} \mathbf{X}_{orthogonal} &= \mathbf{X} - \frac{error}{2}\mathbf{Z} \\ \mathbf{Z}_{orthogonal} &= \mathbf{Z} - \frac{error}{2}\mathbf{X} \end{aligned} \tag{8}$$

The following step is to get the middle row of the rotation matrix. This is simply obtained by taking a cross-product of the two vectors $\mathbf{X}_{orthogonal}$ and $\mathbf{Z}_{orthogonal}$. Finally, the row vectors are normalized to each have a magnitude of one.

2.5 Augmenting virtual data

The final phase of the system is to augment the video stream with some meaningful information from the forestry machine to the operator. In this initial prototype, we decided to show a 3D point cloud of the environment and a simple wireframe model of the forestry crane and the tool. The point cloud data gives the possibility to highlight certain parts of the environment, e.g. a tree to be cut. The model of the forestry crane and the tool could be used to show what kind of motion a machine has planned to do. It also gives so called "X-ray vision" for the operator because part of the view is occluded by the structures of the machine.

The point cloud is collected by the 2D laser scanner attached to the side of the forestry crane so that it scans the environment vertically (see Fig. 1). The data points are constantly updated as new information is available, simultaneously rejecting dynamic objects like the swaying tool. The calculation of the point cloud from the laser scanner measurements is omitted from this work because of the limited length of the paper. The point cloud is visualized by colouring the data points according to distance and height.

The model of the forestry crane is obtained from the instrumentation of the forest machine research platform (Kalmari et al., 2013b). Additionally, the method by Kalmari et al. (2013a) is used to localize the freely hanging tool. It uses an EKF based method with two IMUs. One of the IMUs is attached to the tip of the boom and the other is fixed to the tool itself.

In order to show the virtual points on the image, a coordinate transformation from the virtual coordinate system to the marker coordinate system has to be estimated. In our approach, several undistorted images, where the tip of the boom is visible, are recorded. The images contain different boom locations in a wide area. Together with the images, the pose of the head in marker frame and the location of the boom tip in virtual frame are saved. In Matlab, image coordinates of the boom tip are then acquired by mouse-clicking the target from each image. A 3D vector in marker frame can be estimated using the pose and image point information. Moreover, we seek a transformation matrix that minimizes the sum of the squared distances of the virtual points and the 3D vectors. Matlab's numerical minimization function *fminsearch*, which uses the Nelder-Mead simplex algorithm as presented in (Lagarias et al., 1998), is chosen for the task.

3. EXPERIMENTS & RESULTS

Testing of the prototype system was performed outdoors in bright daylight. The operator worked naturally in the forest machine cabin controlling the forestry crane to different positions. The forest machine itself was in place for the whole time but it swayed slightly due to the motion of the crane. All the measurements were recorded during a test sequence 3 minutes 47 seconds long which enabled it to be played back. Before the test sequence, the locations of the markers were automatically estimated from an offline video that was shot inside the cabin.

One of the augmented video images can be seen in Fig. 2. Although there are seemingly some misalignments with real and virtual objects during the video sequence, the overall look is still satisfying most of the time. The motion of the head inside the cabin is presented in Fig. 3. The largest motion happens about the yaw-angle which represents the heading of the operator's head. Moreover, there was only a

Fig. 2. An augmented image from the video. The crane and the tool are shown in blue together with point cloud data from the lidar.

brief moment (~0.3 s) in the sequence when no markers were detected, and thus the virtual data was not augmented.

The augmentation errors can be identified by comparing the visual appearance to different measurements of the pose. Firstly, jitter is found in the test sequence, which is the noise in the pose estimates. Jitter makes the virtual points seem shaky on the images. Jitter is evaluated in this work by inspecting the angular velocities of pitch-, roll-, and yaw-angles. In Fig. 4, there are two measurement sets of angular velocity of pitch-angle (lateral tilt of the head). One of them is acquired directly from the camera estimate and the other is taken after the camera estimate is sensor fused with the DCM estimate of the IMU. There are clearly identifiable moments in time, when the camera estimate becomes significantly noisy. After sensor fusion with the IMU, the noise is reduced.

Another visible error is noted as up and down swaying of virtual objects. The swaying effect is gone when the orientation is acquired only using vision. Hence, the error is deduced to be in the IMU's DCM estimate. The phenomenon can be also seen in Fig. 5, where IMU's estimate of the roll-angle has higher amplitude than the camera estimate.

The proposed real time system was implemented in C++ on an Intel Core i5-760 processor with four cores operating at 2.8 GHz. The system proved to work in real time as long as the size of the virtualized point cloud was restricted. One augmentation of an image should last no more than ~67 ms which is the time between two adjacent camera frames. In Table 1, the mean time consumption measurements of different modules are seen. Most of the computational time (~30 ms) is spent in the detection of markers. The amount of virtual points was limited to 20 000 in our system. The time usage in pose estimation is almost negligible, although the software level synchronization increases the time in sensor fusion.

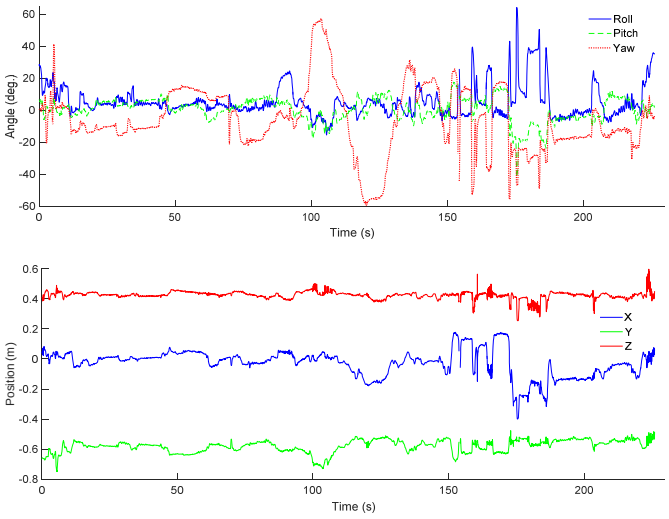

Fig. 3. The position and orientation estimates of the head inside the tractor cabin.

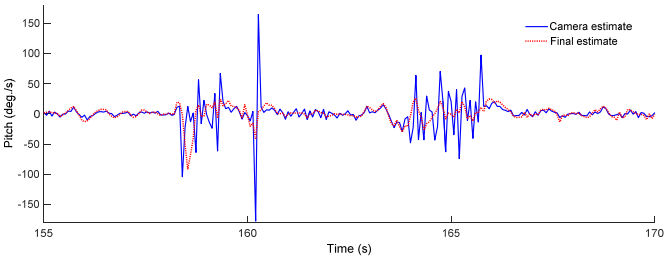

Fig. 4. The angular velocity of pitch-angle used to show jitter.

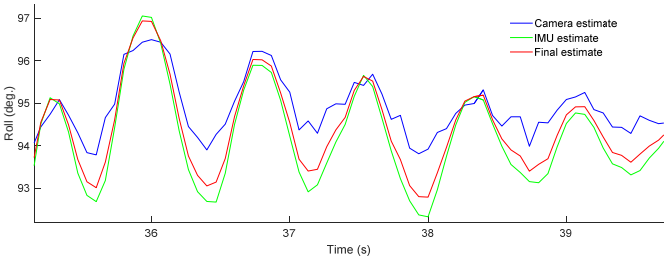

Fig. 5. The amplitude of the IMU's estimated roll-angle is sometimes too high. It is visually shown as up and down swaying of virtual points.

Table 1. Mean time consumption for real time capability.

Phase	**Time**
Marker detection	30 ms
Augmentation	5 ms / 10 000 points
Sensor fusion / synchronization	6 ms
Camera estimate	< 1 ms
IMU estimate	< 1 ms

4. DISCUSSION

The implemented AR prototype proved to work in outdoor environment with real hardware in real time (https://youtu.be/IpHlKVeKzz4). In Fig.2, we can notice that the markers can be attached inside the cabin so that they do not disturb the operator's view of the outside environment. The registration accuracy between the virtual and real points is promising, but there is still development to be made.

One of the problems concerned the vision based pose estimate which became significantly noisy at times (Fig. 4). This was caused by inaccurate detection of markers. Jitter was especially bad, when only one marker was detected on the image with poor detection quality. Sensor fusion with IMU reduces the noise in the orientation estimate. However, the effect stays in the position estimate because it is measured only with the camera.

Another drawback appeared as up and down swaying of the virtual points which was also apparent in Fig. 5. The roll-angle estimate of the IMU had, at times, larger amplitude than desired. The problem exists because the tractor cabin has a suspension system, and the cabin starts to sway due to rapid motions of the forestry crane. The motion of the cabin was not compensated in the sensor fusion module for the IMU's attitude estimate, which measures relative to Earth frame

rather than marker frame. The problem could be solved by attaching an additional IMU on the structures of the cabin.

All other major misalignments were caused by inaccurate detection of some of the markers. This also had a negative effect on the automatic calibration of the marker locations. Whenever the operator had only one of the poorly detected markers in sight, the virtual objects were slightly shifted from their corresponding real objects. The larger the markers appeared on the images the better they were detected. The bright sunlight also reduced the quality of the images which can be seen in Fig. 2.

In the future, the simple sensor fusion module used in this work should be replaced with more comprehensive one. The system could be made more robust by using the IMU also to estimate position. For now, the system cannot measure the position of the head at all when no markers are detected. Additionally, the sensor fusion module could also include online bias correction for accelerometer. E.g., the work by Hol et al. (2007) presents an EKF based sensor fusion method for vision and IMU sensors including these features. Also, Liu et al. (2016) have proposed recently a promising visual-inertial sensor fusion algorithm which could be easily integrated with our design, since position is estimated in a separate filter.

The detection accuracy of the markers is one of the biggest concerns in the proposed system. Because the size of the markers cannot be increased much more, there has to be some other way to improve it. One idea is to illuminate the markers which would sharpen the corners on the images. Also, the selected camera proved to be hampered by sunlight. A potential choice would be to replace it with a CMOS camera with a global shutter.

Furthermore, the system's pose estimation accuracy is to be tested against an accurate reference measurement. Thus, the performance of the pose estimation method could be compared to some previous work. Also, the robustness of the system has to be tested in different lighting conditions and with different kind of user movement. All in all, the methods used in this work have shown potential to be utilized as a part of an augmented reality human machine interface in a forest machine cabin.

5. CONCLUSION

In this paper, we presented the first working real time AR demonstration in a forest machine. The virtual model of the environment and of the forestry crane were built from on-board sensors and augmented on to a live video feed. This setup enables the possibility to show relevant data to the operator intuitively while working with semi-autonomous forest machine. The results suggest that the use of vision with markers and inertial measurements are a valid configuration for pose estimation in a forest machine cabin. A demonstration of the proposed AR interface is available at https://youtu.be/IpHlKVeKzz4.

ACKNOWLEDGEMENTS

Strategic Research Council at the Academy of Finland is acknowledged for financial support of project 'Competence-Based Growth Through Integrated Disruptive Technologies of 3D Digitalization, Robotics, Geospatial Information and Image Processing/Computing – Point Cloud Ecosystem' (COMBAT, project decision number 293389). We would also like to thank Ville Toiviainen for valuable comments on the manuscript.

REFERENCES

Azuma, R.T. (1997). A survey of augmented reality. *Presence,* volume 6 (4), pp. 355-385.

Billingsley, J., Visala, A., and Dunn, M. (2008). Robotics in Agriculture and Forestry. In Siciliano, B. and Khatib, O., *Handbook of Robotics,* pp. 1065-1077. Springer-Verlag.

Bouguet, J.Y. (2004). Camera Calibration Toolbox for Matlab. [online] Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/ [Accessed 31.10.2016].

Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik,* volume 1 (1), pp. 269-271.

Finnish Customs (Tulli) (2015). *Foreign Trade 2014 – Finnish Trade in Figures*.

Food and Agriculture Organization of the United Nations (FAO). (2016). *FAO yearbook of Forest Products 2014*. Rome.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J. and Marín-Jiménez, M.J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition,* volume 47 (6), pp. 2280-2292.

Henderson, S.J. and Feiner, S. (2009). Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. *8th IEEE International Symposium o*n *Mixed and Augmented Reality*, pp. 135-144.

Hol, J.D., Schön, T.B., Luinge, H., Slycke, P.J., and Gustafsson, F. (2007). Robust real-time tracking by fusing measurements from inertial and vision sensors. *Journal of Real-Time Image Processing,volume 2 (2-3), pp. 149-160.*

Hyyti, H. and Visala, A. (2015), A DCM based attitude estimation algorithm for low-cost MEMS IMUs. *International Journal of Navigation and Observation,* volume 2015.

InvenSense. (2010). MPU-6050 MEMS MotionTracking™ Device, PS-MPU-6500A-01 datasheet [revised 19.8. 2013].

Kalmari, J., Hyyti, H. and Visala, A. (2013a). Sway estimation using inertial measurement units for cranes with a rotating tool. *IFAC Proceedings Volumes,* volume 46 (10), pp. 274-279.

Kalmari, J., Pihlajamäki, T., Hyyti, H., Luomaranta, M. and Visala, A. (2013b). ISO 11783 compliant forest crane as a platform for automatic control. *8th IFAC Symposium on Intelligent Autonomous Vehicles,* volume 46 (18), pp. 164-169.

Kalmari, J., Backman, J., and Visala, A. (2014). Nonlinear model predictive control of hydraulic forestry crane with automatic sway damping. *Computers and Electronics in Agriculture*, volume 109, pp. 36-45.

Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. *2nd IEEE and ACM International Workshop on Augmented Reality*, pp. 85-94.

Lagarias, J.C., Reeds, J.A., Wright, M.H. and Wright, P.E. (1998). Convergence properties of the Nelder-Mead simplex method in low dimensions. *Society for Industrial and Applied Mathematics Journal on Optimization,* volume 9 (1), pp. 112-147.

Liu, Y., Xiong, R., Wang, Y., Huang, H., Xie, X., Liu, X., and Zhang, G. (2016). Stereo visual-inertial odometry with multiple Kalman filters ensemble. *IEEE Transactions on Industrial Electronics,* volume 63 (10), pp. 6205-6216.

Markley, F.L., Cheng, Y., Crassidis, J.L. and Oshman, Y. (2007). Averaging quaternions. *Journal of Guidance, Control, and Dynamics,* volume 30 (4), pp. 1193-1197.

Nordlie, A. and Till, S. (2015). Head-mounted displays for harvester operators−a pilot study. Master's thesis, KTH Royal Institute of Technology, Stockholm.

Phuong, N. H. Q., Kang, H.J., Suh, Y.S., and Ro, Y. S. (2009). A DCM based orientation estimation algorithm with an inertial measurement unit and a magnetic compass. *Journal of Universal Computer Science,* volume 15 (4), 859-876.

PJRC. (2013). Teensy USB Development Board. [online] Available at: http://www.pjrc.com/teensy/ [Accessed 2.10.2016].

Premerlani, W. and Bizard, P. (2009). Direction cosine matrix imu: Theory. *DIY DRONE: USA,* pp. 13-15.

Rowberg, J. (2016). I2C Device Library – MPU6050. [online] Available at: https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050 [Accessed 26. 9.2016].

SICK (2008). LMS221 Laser Measurement System, Technical information.

Siltanen, S., Hakkarainen, M. and Honkamaa, P. (2007). Automatic marker field calibration. *In Proceedings of the Virtual Reality International Conference (VRIC*), pp. 261-267.

Tang, A., Owen, C., Biocca, F. and Mou, W. (2003). Comparative effectiveness of augmented reality in object assembly. *Proceedings of the SIGCHI conference on Human factors in computing system*s, pp. 73-80.

VTT Technical Research Centre of Finland. (2009). ALVAR. Available at: http://vtt.fi/multimedia/alvar.html [Accessed 26.9.2016].

Won, S.P. and Golnaraghi, F. (2010). A triaxial accelerometer calibration method using a mathematical model. *IEEE Transactions on Instrumentation and Measurement,* volume 59 (8), pp. 2144-2153.

Zhou, F., Duh, H.B. and Billinghurst, M. (2008). Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. *In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 193-202.