A low cost intelligent localization system based on a camera and an inertial measurement unit (IMU)

O. Casanova [1,2], G.N. Marichal [2], J.L. Gonzalez Mora [1] A. Hernández [2]

[1] Department of Physiology. School of Medicine. University of La Laguna.
[2] Department of System Engineering. School of Industrial Engineering. University of La Laguna, La Laguna, Tenerife 38071, Spain; E-Mails: nicomar@ull.es

Abstract

This paper presents an alternative sensor fusion application in order to improve the accuracy on systems that use strapdown type inertial navigation. A neuro-fuzzy approach is shown, where the information captured by a camera is fused with the values obtained from a low cost inertial measurement unit. Several experimental trials have been conducted with real data, where satisfactory results have been achieved.

Keywords: neural networks; fuzzy systems; inertial sensors; MEMS; inertial navigation

1. Introduction

Current MEMS (micromachined electromechanical systems) inertial sensors are small and lightweight, but suffer some drawbacks that make unreliable a navigation system based only on these devices. Calculating the double integral of the acceleration provided by an IMU (inertial measurement unit) is not a valid method to obtain the position from a known starting point because the MEMS are affected for different noise processes. These noise components are also double integrated causing an error that grows exponentially with respect to the time. Hence, it causes a continuous drift.
Deterministic noise like bias instability is relatively easy to remove, but random noise processes require more advanced techniques. Sensor fusion with local or global positioning systems employing Kalman filters [1] [2] is a common approach to solve the drift problem. However, this technique is not applicable in some cases. Global positioning systems don't work on indoor environments and local positioning systems require complex setup to be functional even on closed environments.
This paper presents an alternative method to determine the position of a moving object employing only an inertial measurement unit (IMU) and a low cost camera. This non-intrusive and autonomous method tries to minimize some of the problems involved on inertial navigation

A description of the subsystems of the proposed localization system will be shown in section 2. Section 3 will be focused on the description of the neuro-fuzzy system used in this approach, while section 4 will be addressed to the application of this particular neuro-fuzzy architecture to the fusion of the information of the different sources. On the other hand, the experimental setup and the carried out trials will be discussed in section 5. Finally, conclusions will be presented in section 6.

2. Global description.

2.1. Mtx IMU sensor

To carry out the experiments, an Xsens MTx IMU sensor was employed. This sensor is a small and good quality MEMS device that includes three gyroscopes and three accelerometers. It is capable of returning the orientation values without external processing. Xsens provides a correction method using sensor fusion to prevent the drift effect in orientation. An own implementation of the Extended Kalman Filter is used by Xsens to combine the angular acceleration provided by the gyroscopes with an internal magnetic compass and gravity vector measurement provided by the accelerometers. The application of this technique results in a very stable orientation information with reasonable drift values along the time.
However, linear acceleration is not susceptible of being corrected easily due to lack of information to combine with by a sensor fusion algorithm.

2.2. Strapdown algorithm

Strapdown algorithm is a classical technique used in navigation. Once the acceleration of a mobile system is known, the fundamental idea of this approach is to integrate this acceleration with respect to the time, obtaining the velocity. Moreover, this velocity is also integrated in order to obtain the position of the mobile system. The algorithm supposes that the initial values of the acceleration, velocity and position are known.
Considering the Xsens MTx IMU with three accelerometers orientated according to the three axes of an orthogonal frame system, it is possible to

measure the Cartesian components of the acceleration vector with respect to a reference system fixed on the device. That is, each accelerometer is positioned in each axis.

Because of that, the acceleration vector obtained by the three accelerometers should be projected over the axes of the world reference system. Once, these projected accelerations are calculated, it is necessary to eliminate the vertical component of the gravity acceleration. In this way, it is possible to obtain the real acceleration, that is, the own acceleration of the mobile object.

Once this real acceleration is calculated, it is possible to obtain the position of the mobile objet with respect to the world reference system. Only, it is necessary to carry out a double integration.

2.3. Error sources

As it was pointed out in previous sections the reliability of the strapdown algorithm is subjected to several error sources. In order to determine the magnitude of the errors a first experience has been carried out. It consists in using the IMU sensor firmly fixed to a table with the local and global frame of reference aligned. Data was captured along 30 seconds at 100 samples per second. This data was double integrated with zero initial position, velocity and acceleration. The sensor showed a drift of 15 meters in only 30 seconds applying the strapdown navigation algorithm.

The error was calculated as the Euclidean distance to the frame origin. Note that, the frame origin coincidences with the initial position in this case.

There are three main error sources involved in drift:

- Bias error. In the strapdown algorithm, the bias is double integrated and causes a quadratic order drift along the time. This error is relatively easy to remove, calculating the bias value for each axis and subtracting them.
- Misalignment between world frame of reference and sensor frame of reference. Minor errors in orientation data cause errors in the gravity vector projection over the world frame of reference. When the gravity acceleration contribution is subtracted, a residual error is presented on each axis due to misalignment errors.
- Thermal noise. White noise caused by electron movements in the electronic components of the IMU sensor

2.4. Vision System

Cameras are low cost and non-invasive devices that can be used to capture movement. The movement can be evaluated measuring the displacement of selected image points in consecutive frames (optical flow). To do that, the Lucas-Kanade algorithm [3] has been used. Lucas-Kanade algorithm is a classic optical flow algorithm usually employed in real time applications due to the good compromise between speed and performance. The points to track are automatically selected looking for corners with big eigenvalues in the image.

The tracking points are distributed along the image in a uniform way. In order to get these tracking points the image is divided into a matrix of cells. In fact n tracking points are chosen for each cell. In this way, it is possible to find a group of tracking points distributed uniformly by the whole image. That is, it is avoided that particular features of different areas of the images as an especial texture area, a high contrast area or others could condition the searching algorithm. Moreover, this division into cells could help to avoid the negative influence of mobile objects. Note that, points associated to mobile objects could be chosen by the algorithm. Unfortunately, it is a bad election given these points will introduce errors in the computation. However, as the points have been considered distributed around the whole image, it is less likely a significant number of points corresponding to mobile objects are chosen. Note that, the optical flow could attend to the movement of the mobile object, instead of the camera movement.

In order to keep a constant number of tracking points in each cell, some of them have been eliminated, whereas others are appended to the cell. When more than n points fall into a single cell (due to point displacements), the exceeding points are discarded. In the same way, when a cell has less than n points, a new one is automatically created into the cell. In the current implementation, a 5x2 grid of cells is used. In fact, only one tracking point for each cell has been chosen for the sake of simplicity. It is important to precise that each cell has one tracking point and this tracking point will be substituted by another if it is not in this cell in the following frame. Furthermore, a new tracking point is searched for this cell. This new point will be chosen among the available tracking point for this cell. In fact, the easiest one in order to be recognized according to Lucas-Kanade algorithm will be chosen.

3. Neuro-Fuzzy system.

The structure of the used Neuro-Fuzzy is similar to the Artificial Network Fuzzy Inference system proposed by Jang. [8].

In this case, only three layers have been considered. The first layer or input layer comprises several nodes, each one consisting of a Radial Basis neuron. The inputs to the radial basis neurons are the inputs to the Neuro-Fuzzy System, while the outputs of the nodes are as follows:

$$p_{ij} = \exp\left(-\frac{\left(U_i - m_{ij}\right)^2}{\sigma_{ij}^{\,2}}\right) \quad \begin{matrix} j = 1,2,\ldots, N2 \\ i = 1,2,\ldots, N1 \end{matrix} \qquad (1)$$

Where:

m_{ij}= Centre of the membership function corresponding to ith input and the jth neuron of the hidden layer.
U_i= ith Input to the Neuro-Fuzzy System.
σ_{ij}= Width of the membership function corresponding to the ith input and the jth neuron of the hidden layer.
p_{ij}= Output of the Radial Basis neuron (or degree of membership for ith input corresponding to jth neuron).
N2 = number of nodes at the hidden layer.
N1 = number of Neuro-Fuzzy System inputs.

On the other hand, the node outputs corresponding to the hidden layer are calculated as:

$$\gamma_j = \min\left[p_{1j}, p_{2j}, \ldots, p_{ij}, \ldots, p_{N1j}\right] \quad j = 1,2,\ldots,N2 \tag{2}$$

Where:

γ_j = Output of the jth node at the hidden layer.

Finally, the output layer could be considered as a linear neuron layer, where the weight connections between the hidden layer and the output layer are the estimated values of the outputs. The outputs of these nodes are calculated by this expression:

$$Y_k = \frac{\sum_j sv_{jk}\gamma_j}{\sum_j \gamma_j} \qquad \begin{matrix} j = 1,2,\ldots,N2 \\ k = 1,2,\ldots,N3 \end{matrix} \tag{3}$$

Where:

Y_k= kth output of the Neuro-Fuzzy System.
sv_{jk} = Estimated value of the kth output provided by jth node at the hidden layer.
N3= number of outputs of the Neuro-Fuzzy System.

Hence, it could be said that the output layer carries out the defuzzification process, providing the outputs of the Neuro-Fuzzy System.
To sum up, the structure of the Neuro-Fuzzy system could be seen as a typical Radial Basis Network, where an additional layer has been inserted between the Radial Basis layer (the input layer) and the linear layer (the output layer). The neurons of this additional layer calculate the degrees of membership corresponding to the different rules. That is, they apply the fuzzy operator AND, being N2 the total number of Fuzzy rules. Once, these calculations have been carried out the output layer applies a defuzzification process in order to obtain numeric values for the outputs of the system.
In this Neuro-Fuzzy system, there are three parameters, which determine mainly the relation between inputs and outputs. These parameters are: the centers of membership functions, the widths of the membership functions and the estimated values of the outputs.
In order to determine these parameters is usual to use a learning algorithm divided into two phases [4] [5][6]. In this paper, it has been used the learning algorithm depicted by Marichal G.N. et al. [7]. The first phase is focused on obtaining initial values for the system parameters. Moreover, an optimization process of the nodes at the hidden layer is carried out. Hence, a reduction of the number of Fuzzy rules is achieved. At this point, an initial zero-order Sugeno Fuzzy System is obtained. On the contrary, the last phase is focused on re-adjusting the different parameters of the Neuro-Fuzzy System, taking as initial values the parameters obtained in the previous phases. This approach is usual in so many learning algorithms associated to Neuro-Fuzzy Systems. In particular, an algorithm based on the Kohonen network has been applied in the first phase and an algorithm based on the least squared method has been applied in the second one.

4. System integration

In order to combine the information of several sources a prototype integrating a camera and the Mtx IMU has been designed. As it can be seen in Fig 1 a cap has been designed where the camera and the Mtx IMU have been incorporated. Additionally, the electronics for capturing the data of both devices has been appended to the prototype. All components have been firmly fixed to a plastic structure. In this way, it is easier to do the experimental trials.
Once, the prototype has been designed it is necessary to devise a strategy to process the information.
In this paper, a Neuro-Fuzzy approach has been used to fuse the information in order to improve the accuracy of the calculated positions. As it was explained in section 2.2, the strapdown algorithm is affected by different error sources introducing certain inaccuracy in the obtained results. In order to improve the results the data provided by a camera and the own IMU data could be fused.
.

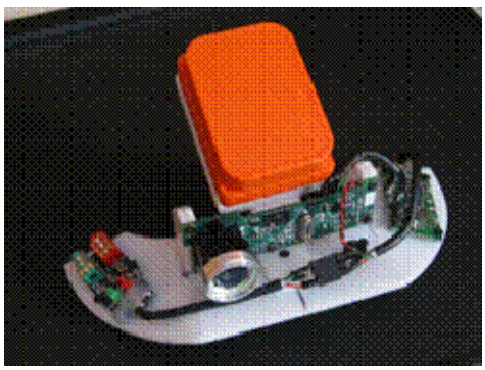

Fig. 1- Prototype.

In this paper, the orientation information, the strapdown algorithm results and the Lucas-Kanade

algorithm tracking points have been used as inputs to the Neuro-Fuzzy system for improving the accuracy of the obtained positions. In particular, the orientation information is expressed by the three angles: roll, pitch and yaw. Each one is used as a different input to the Neuro-Fuzzy system.

On the other side, the strapdown algorithm results are introduced to the Neuro-Fuzzy system as three inputs. Each input corresponds to the calculated Euclidean coordinates with respect to the world reference system. At last, the Lucas-Kanade algorithm tracking points have been expressed as vectors between two consecutive tracking points in two frames. Note that, these tracking points correspond to the same physical points in the scene. In fact, ten vectors have been considered given the image has been divided into ten cells, choosing one tracking point per cell. Moreover, a polar notation has been considered for these vectors. That is, a module and an argument have been considered for each vector.

Taking into account the considerations carried out above a Neuro-Fuzzy with twenty six inputs would be necessary. In Fig. 2 a block diagram of the fusion scheme is shown. It is important to remark that this number of inputs is high for the Adaptive Network Fuzzy Inference System (ANFIS) proposed by Jang [8]. Instead of it, the Neuro-Fuzzy system depicted in section 3 has been used.

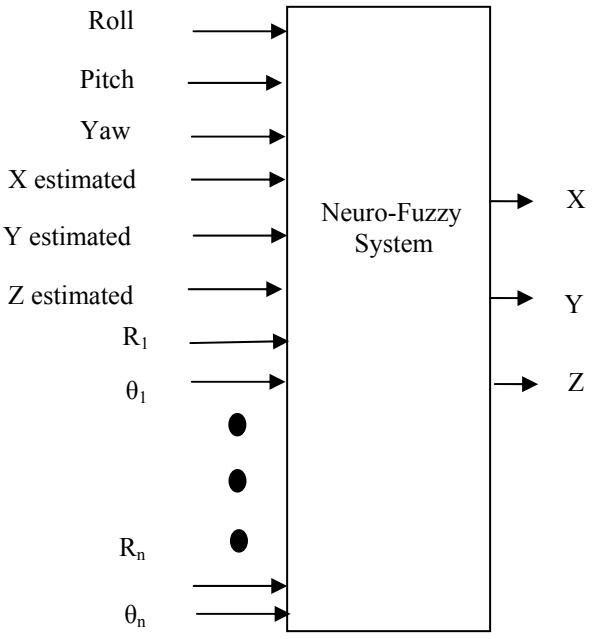

Fig 2- block diagram of the Neuro-Fuzzy system.

5. Experimental setup

In order to carry out the training phase of the Neuro-Fuzzy system it is necessary to have a set of real positions. This set of real positions has been measured by a position measurement system. In this case, the 3D Polhemus Fastrak has been used. This system is able to give position and orientation measurements at a frequency of 120 Hz. The measurement terminal of the Fastrak was fixed to the prototype structure. Furthermore, a computer program was developed to capture the data at the same time. In fact, the following data were captured:

Acceleration provided by the IMU.
Orientation provided by the IMU.
The module and argument of ten vectors associated to the tracking points of the current and previous frame.
Positions in coordinates with respect to the world reference system obtained by the Fastrak system.

In Fig. 3 it is shown a photograph of the work environment. As it can be seen a work environment with several objects has been used in the data acquisition phase. A person with the prototype is moving inside the environment, while the Fastrak system and the developed software is capturing data in real time. It is important to remark that the sensors used in the experiments have different sample rate. In fact, the rate of image capture determined the greatest delay. Because of that, it is necessary to process the obtained data in order to assure the data synchronization.

Fig. 3- Tracker room

In order to overcome this drawback a spline interpolation has been used for predicting the IMU values at the frame rate.

In this way, a set of patterns have been obtained. In fact, 2400 patterns have been considered. Note that, each pattern corresponds to a set of inputs to the neuro-fuzzy system. This set of patterns has been divided into two groups. First group has been used for the training process and the second one has been used to determine the generalization properties of the

resultant fuzzy system. In fact, the seventy percentages of data were used for the leaning phase and the rest of them were used in the generalization tests.

First trials suggested that it is better to use one neuro-fuzzy system for each position coordinate. That is, three neuro-fuzzy systems with a unique output were used.

At the first stage of the algorithm several trials were carried out where it was found a satisfactory sum squared error after 20000 epochs. Once, this unsupervised learning phase was concluded, several new trials were carried out in the supervised phase of the algorithm. In this case, several choices of the learning rates were done in order to achieve a low sum squared error. In particular, two sum squared error curves have been considered. First one was the squared error curve with the training patterns and the other with the squared error curve corresponding to the patterns used for the generalization tests. In this way, it was possible to observe the evolution of the generalization capabilities of the neuro-fuzzy system.

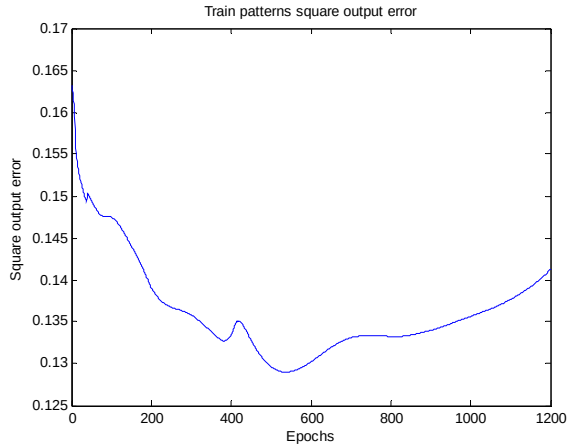

Fig. 4 – sum squared error of the training patterns vs. epochs.

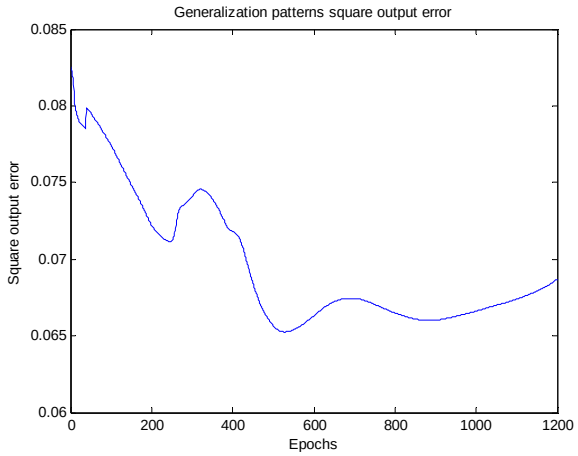

Fig. 5 – sum squared error of the generalization patterns vs. epochs.

In Fig. 4, it is shown the evolution of the sum squared error for the z coordinates of the positions. Fig 4 shows the error curve for the training patterns, whereas Fig. 5 shows the error curve for the patterns used in generalization. It is important to remark, that Fig. 4 shows the evolution of the errors between the outputs of the Neuro-Fuzzy system and the real position values for the set of samples used in the training process. However, Fig. 5 shows the errors for a set of samples which are not used in the training process, taken as test set.

Similar curves have been obtained for y and x coordinates. In the following table, it is shown the minimum squared error for each axis. Note that, these values are extracted from the generalization error curve as the curve shown in Fig 5 for the z coordinates. Because of that, the parameter values for the final neuro-fuzzy system will be chosen at this epoch of the training process. It is important to remark that once the training process has been finished the neuro-fuzzy system with its own fixed parameters is equivalent to a zero- order Sugeno fuzzy system.

Axis	Epoch	Generalization error
X axis	4637	0,067
Y axis	283	0,329
Z axis	528	0,065

Table 1: minimum generalization error.

In Fig 6, it is shown the y-coordinate curve obtained by the neuro-fuzzy approach and the measured data by the fastrak versus time. In this case, it could be seen that a maximum error of 150 cm have been obtained.

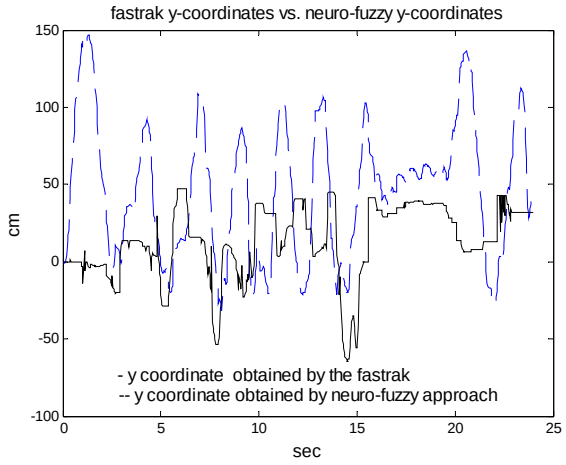

Fig. 6 – y-coordinates obtained by neuro-fuzzy approach vs. y-coordinates obtained by fastrak.

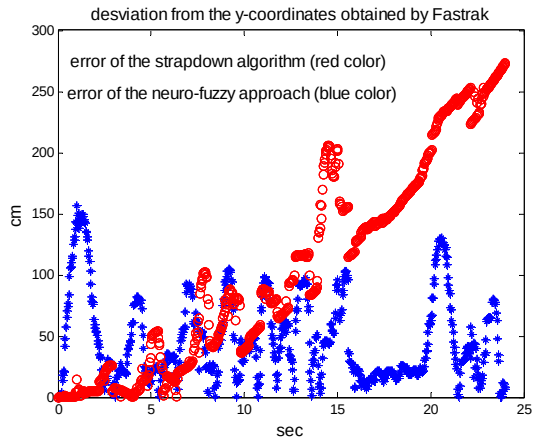

Fig 7 – desviation of the y-coordinates

In Fig.7, it is shown the error values corresponding to the neuro-fuzzy approach versus the strapdown algorithm errors. As it can be seen strapdown algorithm achieves small errors at the beginning, however after 12 seconds the errors increased notably. Whereas, the neuro-fuzzy approach keep a more stable error behavior. In fact, if it were considered the mean for each second, it could be seen that the eighty-seven percentage of neuro-fuzzy approach errors are situated below 60 cm, however only the forty percentage of the errors are situated below 60 cm in the strapdown algorithm case. Moreover, the maximum error for the neuro-fuzzy approach is 118 cm, whereas a maximum error of 243 cm is achieved for the strapdown algorithm in the considered time window of 25 seconds. It is important to remark that greater errors will be achieved with the strapdown algorithm when more elapsed time is considered.

6. Conclusions

In this paper a strategy of combining the information provided by a MEMS IMU and the information extracted from the images captured by a camera have been presented. Furthermore, this strategy has been based on the application of a neuro-fuzzy approach. In fact, a set of experimental data have been collected from a real scenario in order to test the algorithm. Several trials have been carried out using the neuro-fuzzy approach, obtaining satisfactory results. Moreover, a typical strapdown algorithm has also applied. Finally, the results have been discussed using generalization data.

Acknowledgements

This work has been financed by the Spanish government projects TIN2008-06867-C02-01 of Ministerio de Industria, Turismo y Comercio, TSI-020100-2009-541 and DPI2010-20751-C02-02 of Ministerio de Ciencia e Innovación.

References

[1] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, “Circumventing dynamic modeling: Evaluation of the error-state Kalman filter applied to mobile robot localization,” in Proc. IEEE Int. Conf. Robot. Autom., Detroit, MI, May 10–15, 1999, vol. 2, pp. 1656–1663.

[2] D. Rodriguez-Losada, F. Matia, L. Pedraza, A. Jimenez and R. Galan. "Consistency of SLAM-EKF Algorihtms for Indoor Environments", Journal of Intelligent and Robotic Systems, Vol. 50. 2007, pp. 375-397.

[3] B. D. Lucas and T. Kanade (1981), An iterative image registration technique with an application to stereo vision. Proceedings of Imaging Understanding Workshop, pages 121--130

[4] Chen C.H., (1996) .” Fuzzy Logic and Neural Networks Handbook”, Mc. Graw-Hill.S.

[5] Mitra, Y. Hayashi, (2000). Neuro-fuzzy rule generation: Survey in soft computing framework, IEEE Transactions on Neural Networks 11 (3) 748-768.

[6] B. M. Al-Hadithi, F. Matía, A. Jiménez. 2007. Analysis of Stability of Fuzzy Systems. Revista Iberoamerica de Automática e Informática Industrial. Vol. 4. pp. 7-25.

[7] Marichal G. N., Acosta L., Moreno L., Méndez, J.A., Rodrigo J.J., Sigue M.. (2001) “Obstacle avoidance for a mobile robot: A neuro-fuzzy approach”. Fuzzy Set and Systems. Vol. 124, Nº 2, pp. 171-179.

[8] J.-S. R. Jang, ANFIS: Adaptive-Network-based Fuzzy Inference Systems, IEEE Trans. on Systems, Man, and Cybernetics, vol. 23, pp. 665-685, May 1993.

[9] Woodman Oliver J. An introduction to inertial navigation. Technical Report. - Cambridge: University of Cambridge, 2007. - ISSN 1476-2986.

[10] S.-H. Jung and C.J. Taylor. Camera trajectory Estmation using inertial sensor measurements and structure from motion results. In Proceedings ofthe IEEE Computer Society Conference on Computer-Vision and Pattern Recognition, volume 2, pp.II–732–II–737, 2001.

[11] M. V. Srinivasan, M. Poteser, K. Kral, (1999).Motion detection in insect orientation and navigation, Vision Research 39 pp. 2749-2766.

[12] A. Si, M. V. Srinivasan, S. Zhang, (2003) Honeybee navigation: Properties of the visually driven odometer, The Journal of Experimental Biology 1265-1273.

[13] L. Muratet, S. Doncieux, Y. Briere, J.-A. Meyer, (2005) A contribution to vision-based autonomous helicopter flight in urban environments, Robotics and Autonomous Systems (Elsevier) 50 (4) 195-209.

[14] S. Hrabar, G. S.Sukhatme, 2004,A comparison of two camera con¯gurations for optic-flow based navigation of a UAV through urban canyons, in: Proc. of the IEEE International Conference on Intelligent Robots and Systems, Japan, pp. 2673-2680.

[15] A. Azarbayejani, A. Pentland, (1995) Recursive estimation of motion, structure, and focal

length, IEEE Trans. Pattern Analysis and Machine Intelligence 17 (6) 562-575.

[16] M. Wei and K. P. Schwarz, “Testing a decentralized filter for GPS/INS integration,” in Proc. IEEE Plans Position Location Navig. Symp., Mar. 1990, pp. 429–435.

[17] S. Feng and C. L. Law, “Assisted GPS and its impact on navigation transportation systems,” in Proc. 5th IEEE Int. Conf. ITS, Singapore, Sep. 3–6, 2002, pp. 926–931.

[18] D. Bevly, J. Ryu, and J. C. Gerdes, “Integrating INS sensors with GPS measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness,” IEEE Trans. Intell. Transp. Syst., vol. 7, no. 4, pp. 483–493, Dec. 2006.

[19] R. Gregor, M. Lutzeler, M. Pellkofer, K. H. Siedersberger, and E. D. Dickmanns, “EMS-Vision: A perceptual system for autonomous vehicles,” IEEE Trans. Intell. Transp. Syst., vol. 3, no. 1, pp. 48–59, Mar. 2002.

[20] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, “Stereo vision-based vehicle detection,” in Proc. IEEE Intell. Veh. Symp., 2000, pp. 39–44.

[21] A. D. Sappa, F. Dornaika, D. Ponsa, D. Geronimo, and A. Lopez, “An efficient approach to onboard stereo vision system POSE estimation,” IEEE Trans. Intell. Transp. Syst., vol. 9, no. 3, pp. 476–490, Sep. 2008.