IEEE International Workshop on Imaging
Systems and Techniques – IST 2007
Krakow, Poland, May 4–5, 2007

A Rotational and Translational Image Stabilization System for Remotely Operated Robots

A. Amanatiadis[1], I. Andreadis[1], A. Gasteratos[2], N. Kyriakoulis[2]

[1]Laboratory of Electronics, Department of Electrical and Computer Engineering,
Democritus University of Thrace, 12 Vas. Sofias Str, GR-67100, Xanthi, Greece
E–mail: {aamanat, iandread} @ee.duth.gr

[2]Laboratory of Robotics and Automation, Department of Production and Management Engineering,
Democritus University of Thrace, University Campus Kimmeria, GR-67100, Xanthi, Greece
E–mail: {agaster, nkyriako}@pme.duth.gr

Abstract – *Remotely operated robots equipped with on board cameras, apart from providing video input to operators, perform optical measurements to assist their navigation as well. Such image processing algorithms require image sequences, free of high frequency unwanted movements, in order to generate their optimal results. Image stabilization is the process which removes the undesirable position fluctuations of a video sequence improving, therefore, its visual quality. In this paper, we introduce the implementation of an image stabilization system that utilizes input from an on board camera and a gyrosensor. The frame sequence is processed by an optic flow algorithm and the inertial data is processed by a discrete Kalman filter. The compensation is performed using two servo motors for the pan and tilt movements and frame shifting for the vertical and horizontal movements. Experimental results of the robot head, have shown fine stabilized image sequences and a system capable of processing* 320×240 *pixel image sequences at approximately* $10\ frames/sec$*, with a maximum acceleration of* $4\ deg/sec^2$.

Keywords – *Image stabilization, visuo-inertial model, robot navigation.*

I. INTRODUCTION

Recent demand in sophisticated mobile robots require many semi-autonomous or even autonomous operations, such as decision making, simultaneous localization and mapping, motion tracking and risk assessment, when operate in dynamic environments. Most of these capabilities highly depend on the quality of the input from the cameras mounted on the mobile platforms and require fast processing times and responses. However, quality in vision systems is not given only by the quantitative features such as the resolution of the cameras, the frame rate or the sensor gain, but also by the qualitative features such as sequences free of unwanted movement, fast and good image pre-processing algorithms and real-time response. A system having optimal quantitative features for its vision system cannot achieve the finest performance when the qualitative features are not met. Image stabilization is one of the most important qualitative features for a mobile robot vision system, since it removes the unwanted motion from the frame sequences captured from the cameras. This image sequence enhancement is necessary in order to improve the performance of the subsequently complicated image processing algorithms that will be executed. Several stabilization system implementations that use visual and inertial information have been reported. An image stabilization system which compensates the walking oscillations of a biped robot is described in [1]. A vision and inertial cooperation for stabilization have been also presented in [2] using a fusion model for the vertical reference provided by the inertial sensor and vanishing points from images. A visuo-inertial stabilization for space variant binocular systems has been also developed in [3], where an inertial device measures angular velocities and linear accelerations, while image geometry facilitates the computation of first-order motion parameters. In [4], course stabilization and collision avoidance is achieved using a bioinspired model of optic flow and inertial information applied to autonomous flying robots.

In this paper, we present a rotational and translational image stabilization system for a pan and tilt stereo camera head. The system is designed to fit into mobile rover platforms allowing the architecture to be modular and the whole system expandable. Special attention was paid to the real-time constraints, particularly for the control part of the system. The stabilization system as shown in Fig.1, consists of a Eurohead [5], two high resolution digital cameras, a DSP inertial sensor, four actuators and controllers and two processing units. Pan and tilt compensation is achieved though mechanical servoing while vertical and horizontal compensation is achieved by frame shifting through a digital frame stabilization algorithm. A key feature is the real-time servo control system, written in C, using Open Source Software which includes a Linux-based Real-Time Operating System, a Universal Serial Bus to RS-232 serial driver, CAN bus drivers and an open source network communication protocol for the communication between the two processing units.

1-4244-0965-9/07/$25.00 ©2007 IEEE

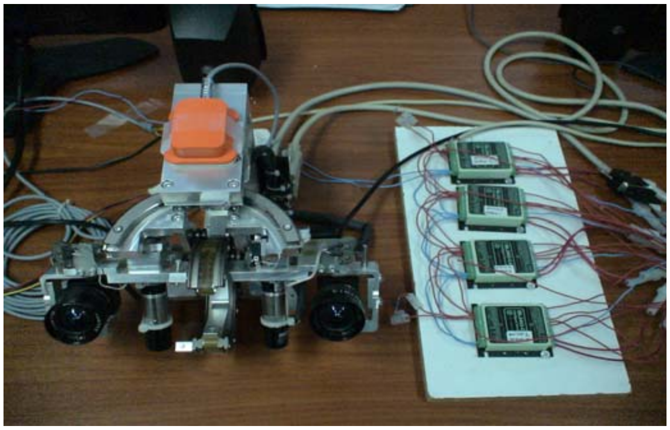

Fig. 1. The stereo head vision system. The modules shown here are: the Eurohead with the fixed actuators, the two digital cameras, the inertial sensor and the four controllers.

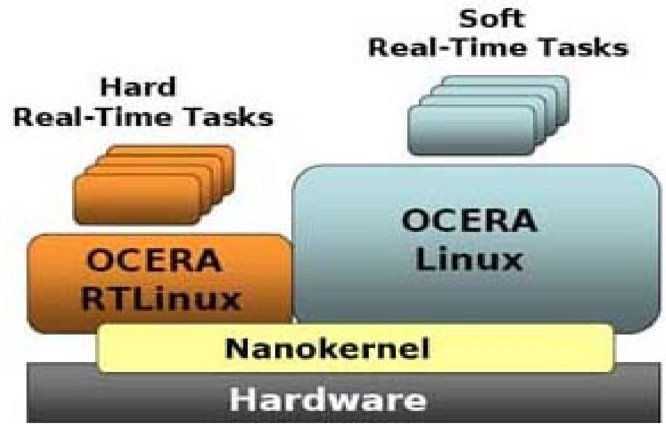

Fig. 2. The chosen operating system combines the use of two kernels, Linux and RTLinux-GPL to provide support for critical tasks (RTLinux-GPL executive) and soft real-time applications (Linux kernel).

II. PROPOSED ARCHITECTURE

A. Hardware Architecture

The system functions can be separated into information processing and motion control. Information processing includes the gyrosensor output and the image processing. However, image processing presents a high computational burden and recourses while it demands the full usage of certain instruction sets of a modern microprocessor. In contrast, motion control requires the operation system to be able to execute real-time tasks. This demand for high multimedia performance and real-time motion control has forced us to adopt a computer structure consisting of a computer with Windows operating system for the image processing and a computer with RT-Linux operating system for the control tasks. The computers are connected to each other by a high speed network protocol for synchronization and frame compensation purposes. This inter-host communication protocol between the computers uses a higher level abstraction, built on top of sockets, meeting the requirements for low latency. The interfaces used are CAN bus for the controllers [6], USB 2.0 for the cameras a USB 1.0 to serial output for the inertial sensor. The drivers for the interfaces connected to the RT-Linux computer are open source under the General Public License.

In order to fully utilize the advantages and precision of the modern digital servo drives a fine tuning process [7] for the pan and tilt PID controllers was carried out. The tuning was orientated for position control and due to the different inertia load seen on the motor shaft of the pan and tilt axis the integral, derivative and proportional band values were set to different values for each axis respectively. In order to determine the internal camera geometric and optical characteristics, camera calibration was necessary. A variety of methods have been reported in the bibliography. The method we used is described in [8] using its available C Open Source code. The method is a non self-calibrating thus, we used a projected chessboard pattern to estimate the camera intrinsics and plane poses. Finally, the calibration results were used to rectify the images taken from cameras in order to have the best results in the subsequent image processing algorithms.

B. Software Architecture

Key feature for the implementation of the real-time control is the operating system we used. Since critical applications such as control, need low response times, OCERA operating system [9] was chosen. OCERA is an Open Source project which provides an integrated execution environment for embedded real-time applications. It is based on components and incorporates the latest techniques for building embedded systems. OCERA architecture is designed to develop hybrid systems with hard and soft real-time activities as shown in Fig.2. In this case, we allocated the critical task of control at the RTLinux level and the less critical tasks, such as inertial data filtering, at the Linux level. The interface for both kinds of activities is a POSIX based interface.

For motion estimation, the rectified frames are processed with an optic flow method in order to extract the global motion translation vector for the motion compensation. The affine model of the optic flow that was used is described in [10] for the basis of frame translation, using a single camera input. For motion compensation process, the estimation method in [11] was selected, in order to remove the undesired shaking motion and simultaneously maintain the ego-motion of the stereo head.

The digital inertial sensor consists of a compact sensor package, which includes accelerometers and gyros to measure accelerations and angular rates. The errors in the force measurements introduced by accelerometers and the errors in the measurement of angular change in orientation with respect to the inertial space introduced by gyroscopes are two fundamental error sources which affect the error behavior of the rotational stabilization. Furthermore, inertial measurements are corrupted by additive noise [12]. The Kalman filter [13], [14] was used which is a form of optimal estimator, characterized by recursive evaluation using an estimated internal model of the dynamics of the system. The filtering is implemented on the RT-Linux computer where the inertial sensor is attached. Finally, the optimized filter outputs of pan and tilt are the subsequent feedback to the controllers for opposite movement of the pan and tilt axis, respectively.

III. ALGORITHM IMPLEMENTATION

A. Kalman Filtering

Discrete Kalman filter computes the best estimate of the systems's state at t_k, $\bar{x}$, taking into account the state estimated by the system model at t_{k-1} and the measurement, z_k, taken at t_k. The Kalman filter equations are characterized by the state covariance matrices, P_k and $P_k^{'}$, and the gain matrix, K_k. $P_k^{'}$ is the covariance matrix of the k-th state estimate

$$\bar{x}_k^{'} = \Phi_{k-1}\bar{x}_{k-1} \tag{1}$$

predicted by the filter immediately before obtaining the measurement z_k, where Φ_{k-1} is a time dependent $n \times n$ matrix called state transition matrix. P_k is the covariance matrix of the k-th state estimate, $\bar{x}_k$ computed by the filter after integrating the measurement, z_k, with the prediction, $\bar{x}_k^{'}$. The covariance matrices are a quantitative model of the uncertainty of $x_k^{'}$ and x_k. Finally, K_k establishes the relative importance of the prediction, $\bar{x}_k^{'}$, and the state measurement, $\bar{x}_k$. Let Q_k and R_k be the covariance matrices of the white, zero-mean, Gaussian system and measurement noise respectively. The Kalman filter equations are

$$P_k^{'} = \Phi_{k-1}P_{k-1}\Phi_{k-1}^{\top} + Q_{k-1} \tag{2}$$

$$K_k = P_k^{'}H_k^{\top}(H_kP_k^{'}H_k^{\top} + R_k)^{-1} \tag{3}$$

$$\bar{x}_k = \Phi_{k-1}\bar{x}_{k-1} + K_k(z_k - H_k\Phi_{k-1}\bar{x}_{k-1}) \tag{4}$$

$$P_k = (I - K_k)P_k^{'}(I - K_k)^{\top} + K_kR_kK_k^{\top} \tag{5}$$

Using (2) to (5), we estimate the state and its covariance recursively. Initial estimates of the covariance matrix P_0 and of the state, $\bar{x}_0$, were set to 0 and 1 respectively [13]. First, $P_k^{'}$ is estimated according to (2). Second, the gain of the Kalman filter is computed by (3), before reading the new inertial measurements. Third, the optimal state estimate at time t_k, $\bar{x}_k$, is formed by (4), which integrates the state predicted by the system model ($\Phi_{k-1}\bar{x}_{k-1}$) with the discrepancy of prediction and observation ($z_k - H_k\Phi_{k-1}\bar{x}_{k-1}$) in a sum weighted by the gain matrix, K_k. Finally, the new state covariance matrix, P_k, is evaluated through (5). In our inertial sensor, the calibrated rate of turn noise density is $0.1units/\sqrt{Hz}$ with units in deg/s. Operating in $40Hz$ bandwidth, the noise is $0.015deg/s$. An experiment was carried out to quantify the filtering behavior of the system in real-time. We applied a recursive motion profile to the tilt axis with constant velocity of $0.3deg/sec$. During the experiment the following parameters were stored to estimate the overall performance: (i) the velocity stimulus input reference (ii) position angle of the controlled tilt servo encoder, (iii) output of the inertial sensor and (iv) the Kalman filter output. Fig.3 shows the sample data recorded during the test session. As it can seen the Kalman filtered output is close to the input reference by estimating the process state at a time interval and obtaining feedback in the form of the noisy inertial sensor measurement.

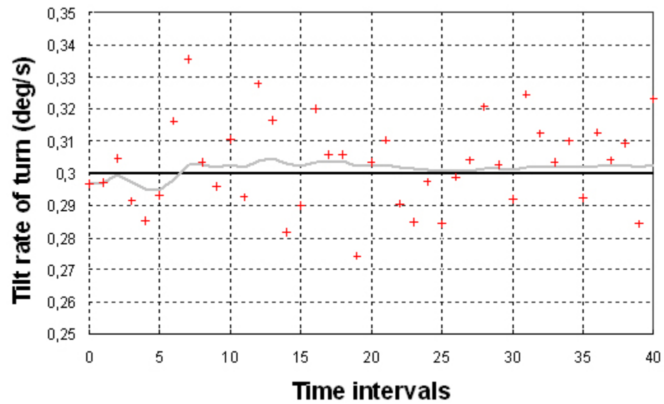

Fig. 3. Sample data during the experiment session. The input reference is $0.3deg/ses$ (black line), the output of the inertial sensor (crosses) and the filtered Kalman output (gray line).

B. Optic Flow and Motion Compensation

Techniques for estimating the motion field are divided in two major classes: differential techniques [15] and matching techniques [16]. A widely used differential algorithm [17] that gives good results was chosen for implementation. Given the assumptions of the image brightness constancy equation yields a good approximation of the normal component of the motion filed and that motion field is well approximated by a constant vector field within any small patch of the image plane, for each point p_i within a small, $n \times n$ patch, Q, we derive

$$(\nabla E)^{\top}\mathbf{v} + E_t = 0 \tag{6}$$

where spatial and temporal derivatives of the image brightness are computed at $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{N^2}$, with $E = E(x, y, t)$ the image brightness and $\mathbf{v}$, the motion filed. Therefore, the optical flow can be estimated within Q as the constant vector, $\bar{\mathbf{v}}$, that minimizes the functional

$$\Psi[\mathbf{v}] = \sum_{\mathbf{p}_i \in Q}[(\nabla E)^{\top}\mathbf{v} + E_t]^2 \tag{7}$$

The solution to this least squares problem can be found by solving the linear system

$$A^{\top}A\mathbf{v} = A^{\top}\mathbf{b} \tag{8}$$

The i-th row of the $N^2 \times 2$ matrix A is the spatial image gradient evaluated at point $\mathbf{p}_i$

$$A = [\nabla E(\mathbf{p}_1), \nabla E(\mathbf{p}_2), \ldots, \nabla E(\mathbf{p}_{N \times N})]^{\top} \tag{9}$$

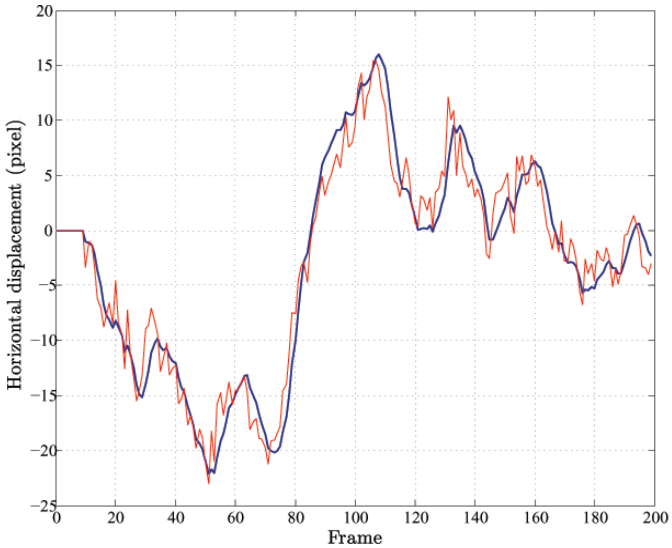

Fig. 4. Horizontal frame sample data during the experiment session. The input before stabilization (red line) and the output after stabilization (blue line) is demonstrated.

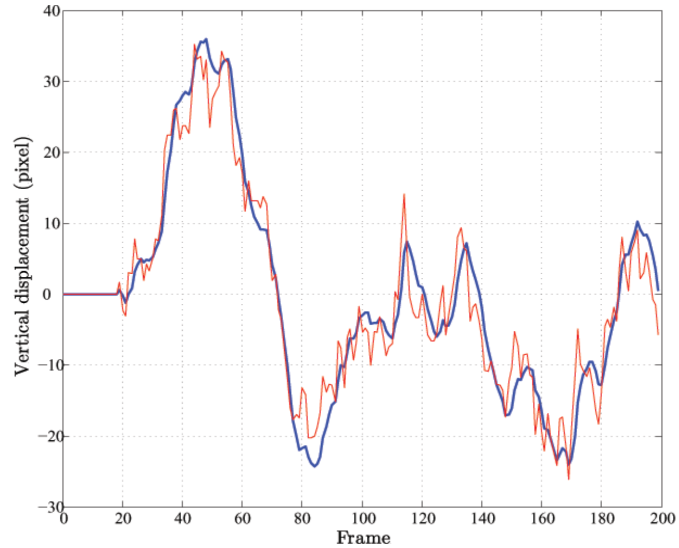

Fig. 5. Vertical frame sample data during the experiment session. The input before stabilization (red line) and the output after stabilization (blue line) is demonstrated.

and $\mathbf{b}$ is the N^2-dimensional vector of the partial temporal derivatives of the image brightness, evaluated at $\mathbf{p}_1, \mathbf{p}_2 \ldots \mathbf{p}_{N^2}$, after a sign change

$$\mathbf{b} = -[E_t(\mathbf{p}_1), E_t(\mathbf{p}_2), \ldots, E_t(\mathbf{p}_{N\times N})]^\top \tag{10}$$

Finally, the optic flow $\bar{\mathbf{v}}$ at the center of patch Q can be obtained as

$$\bar{\mathbf{v}} = (A^\top A)^{-1} A^\top \mathbf{b} \tag{11}$$

Furthermore, we applied to each captured rectified image a Gaussian filter with a standard deviation of $\sigma_s = 1.5$. The filtering was both spatial and temporal in order to attenuate noise in the estimation of the spatial image gradient and prevent aliasing in the time domain. The patch used is 5×5 pixels and three consecutive frames are the temporal dimension. The algorithm is applied for each patch and only the optic flow for the pixel at the center of the patch is computed, generating a sparse motion field with high performance speed of $10 frames/sec$ for 320×240 image resolution.

The Global Motion Vector (GMV) is represented by the arithmetic mean of the local motion vectors in each of the patches and can be potentially effective when subtracting the ego-motion commands of the stereo head which are available through the servo encoders. Subsequently, the compensation motion vector estimation is used to generate the Compensating Motion Vectors (CMVs) for removing the undesired shaking motion but still keeping the steady motion of the image. The compensation motion vector estimation for the final frame shifting is given by [18]

$$\begin{aligned} CMV(t) &= k\,(CMV(t-1)) \\ &\quad + (a\,GMV(t) + (1-a)\,GMV(t-1)) \end{aligned} \tag{12}$$

where t represents the frame number, $0 \leq a \leq 1$ and k is a proportional factor for designating the weight between current frame stabilization and ego-motion. Finally, frame shifting is applied when both horizontal and vertical CMVs are determined.

IV. EXPERIMENTAL RESULTS

Due to the fact that rotational stabilization process runs on the RT-Linux computer we have succeeded its real-time operation. Thus, stabilization can be considered as two separate processes that operate independently, since the frame sequences captured from the camera have been already rotationally stabilized by the mechanical servoing. The horizontal and vertical stabilization experiments are demonstrated in Fig.4 and Fig.5, respectively. The results show a frame sequence free of high frequency fluctuations, maintaining though, the ego-motion of the trajectory. The overall system is capable of processing 320×240 pixel image sequences at approximately $10\, frames/sec$, with a maximum acceleration of $4\, deg/sec^2$.

V. CONCLUSION

A new translational and rotational image stabilization system is proposed which employs a motion estimation optic flow model and an inertial model based on Kalman filtering method. The compensation is performed using two servo motors for the pan and tilt movements and frame shifting for the vertical and horizontal movements. The experimental system running on two separate processing units is effective in suppressing the undesirable jiggling motions. Inertial data processing and control have been implemented in a Linux-based Real-Time Operating System.

ACKNOWLEDGMENT

This work was supported by the E.C. under the FP6 research project for improvement of the emergency risk management through secure mobile mechatronic support for bomb disposal, "RESCUER", IST-2003-511492.

REFERENCES

[1] R. Kurazume and S. Hirose, "Development of image stabilization system for remote operation of walking robots," *in Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1856–1861, 2000.

[2] J. Lobo and J. Dias, "Vision and inertial sensor cooperation using gravity as a vertical reference," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1597–1608, 2003.

[3] F. Panerai, G. Metta, and G. Sandini, "Visuo-inertial stabilization in space-variant binocular systems," *Robotics and Autonomous Systems*, vol. 30, no. 1-2, pp. 195–214, 2000.

[4] J. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Trans. on Robotics and Automation*, vol. 22, no. 1, pp. 137–146, 2006.

[5] A. Gasteratos and G. Sandini, "On the accuracy of the Eurohead," *Lira-lab technical report, LIRA-TR*, vol. 2, 2001.

[6] International Users Manufacturers Group, *Canopen Application Layer and Communication Profile*. Can in Automation, Draft Standard 301, Revision 4.01, 2000.

[7] K. Astrom and T. Hagglund, *PID controllers: Theory, Design and Tuning*. Instrument Society of America, Research Triangle Park, 1995.

[8] J. Bouget, "Camera calibration toolbox for Matlab," *California Institute of Technology, http//www.vision.caltech.edu*, 2001.

[9] OCERA project home page, *http//www.ocera.org*.

[10] J. Koenderink and A. van Doorn, "Affine structure from motion," *Journal of the Optical Society of America*, vol. 8, no. 2, pp. 377–385, 1991.

[11] S. Hsu, S. Liang, and C. Lin, "A robust digital image stabilization technique based on inverse triangle method and background detection," *IEEE Trans. on Consumer Electronics*, vol. 51, no. 2, pp. 335–345, 2005.

[12] S. Ovaska and S. Valiviita, "Angular acceleration measurement: A review," *IEEE Trans. on Instrumentation and Measurement*, vol. 47, no. 5, pp. 1211–1217, 1998.

[13] G. Welch and G. Bishop, "An introduction to the Kalman filter," *ACM SIGGRAPH 2001 Course Notes*, 2001.

[14] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1998.

[15] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[16] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[17] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. DARPA Image Understanding Workshop*, vol. 121, p. 130, 1981.

[18] J. Paik, Y. Park, D. Kim, and S. Co, "An adaptive motion decision system for digital image stabilizer based on edge pattern matching," *IEEE Trans. on Consumer Electronics*, vol. 38, no. 3, pp. 607–616, 1992.