Active gaze stabilization

Martin Lesmana
Computer Science
U. British Columbia
martinlesmana@gmail.com

Axel Landgren
Electrical Engineering
Linköping University
axel.landgren@gmail.com

Per-Erik Forssén
Electrical Engineering
Linköping University
perfo@isy.liu.se

Dinesh K. Pai*
Computer Science
U. British Columbia
pai@cs.ubc.ca

ABSTRACT

We describe a system for active stabilization of cameras mounted on highly dynamic robots. To focus on careful performance evaluation of the stabilization algorithm, we use a camera mounted on a robotic test platform that can have unknown perturbations in the horizontal plane, a commonly occurring scenario in mobile robotics. We show that the camera can be effectively stabilized using an inertial sensor and a single additional motor, without a joint position sensor. The algorithm uses an adaptive controller based on a model of the vertebrate Cerebellum for velocity stabilization, with additional drift correction. We have also developed a resolution adaptive retinal slip algorithm that is robust to motion blur.

We evaluated the performance quantitatively using another high speed robot to generate repeatable sequences of large and fast movements that a gaze stabilization system can attempt to counteract. Thanks to the high-accuracy repeatability, we can make a fair comparison of algorithms for gaze stabilization. We show that the resulting system can reduce camera image motion to about one pixel per frame on average even when the platform is rotated at 200 degrees per second. As a practical application, we also demonstrate how the common task of face detection benefits from active gaze stabilization.

Keywords

gaze stabilization, active vision, Cerebellum, VOR, adaptive control

1. INTRODUCTION

Vision systems in robots, like those in animals, have to function in a dynamic environment. Motion of either the imaged objects or the camera itself causes two significant problems: (1) motion blur, which degrades vision, and (2) disappearance of objects from the field of view, which makes vision impossible. In practice, these are often dealt with using quick fixes: To avoid camera motion blur, the camera motion is restricted. To avoid object motion blur, strong illumination and short shutter speeds are used (alternatively the resolution is reduced). To prevent objects from moving out of view, wide field of view optics are used (with an associated loss of spatial resolution).

In contrast, biological systems deal with fast motions by active movements of the eye to counteract the unwanted motions and keep the image stable on the retina. This *active gaze stabilization* relies on measuring both the acceleration of the head using the vestibular system and image motion in the form of a *retinal slip* signal. Such a stabilization system could, for example, allow a humanoid to recognize objects while walking, or allow a visual SLAM system in a car to work while driving in rough terrain (by suppressing motion blur from bumps and vibrations).

Following the terminology in biology, we will use the word "head" to refer to the platform on which both an actuated camera and an inertial measurement unit (IMU) are mounted. Figure 1a shows the results when the head is subjected to a high speed rotation; figure 1b shows the same motion of the head, but using our system to stabilize gaze.

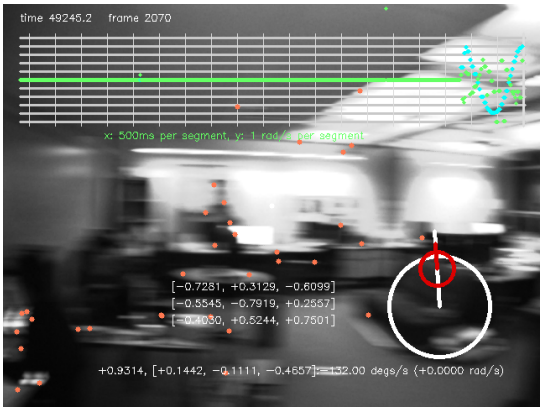
In this paper we describe a complete system for active stabilization of cameras which can reject large disturbances and maintain drift-free object tracking. This is achieved by using both inertial measurement and vision information as inputs to the controller. The controller is adaptive and does not require a system identification step or extensive parameter tuning. It only requires a rudimentary model of the plant and we demonstrate its ability to operate even with a factor of two error in the given plant model DC gain. Even under this condition, the system demonstrates rapid adaptation to a good performance. This robustness is made possible, in part, by the robustness of the vision algorithm used to estimate the retinal slip, which applies crosschecking and resolution adaptivity to a standard feature tracking algorithm. We also describe a robust method for converting motion in the image plane into an angular velocity.

A major goal of this paper is to carefully evaluate the stabilization performance and its limits. While many systems for gaze stabilization have been reported, both in the literature and by commercial vendors, most previous work has only reported examples of their system's performance. In contrast, our system's stabilization performance was evaluated using a high speed robotic platform that can generate repeatable and general rigid motions of the head in the horizontal plane.

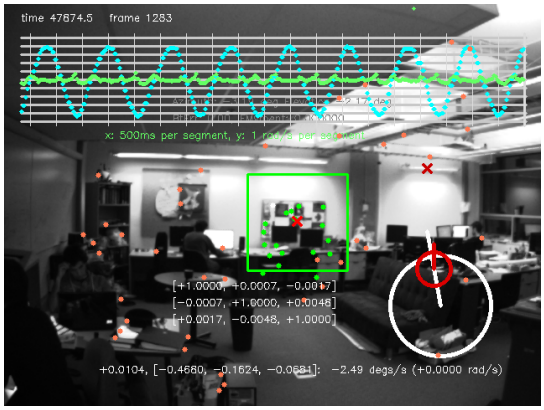
*Corresponding author

ICVGIP '14, December 14-18, 2014, Bangalore, India

http://dx.doi.org/10.1145/2683483.2683565.



(a) Without stabilization



(b) With our stabilization method

Figure 1: Image frames captured by the camera at 60 fps. (a) shows a frame captured near the peak velocity (230°/sec) of a sinusoidal disturbance without stabilization. (b) shows a corresponding frame captured with gaze-stabilization turned on.

2. RELATED WORK

Most of the techniques used for active gaze stabilization, also referred to as image or video stabilization, are targeted at small motions. A common application is suppression of vibrations of hand-held cameras, e.g. by moving the internal camera optics in response to the sensed inertial motion [4, 20]. In [10], an onboard camera is stabilized against vehicle vibration by panning and tilting either a set of external mirrors or the camera itself. Alternatively, one can capture the image as is (without stabilization) and use digital post-processing to correct the motion blurred image [27, 12]. This has been demonstrated to be effective in stabilizing video sequences captured from moving vehicles [15, 18].

With the exception of [10], the above methods are not suited to compensating for large motions encountered, for example, in walking. Consequently, humanoid robots often rely on the larger range of whole camera motions to perform gaze stabilization [3, 24, 21, 8, 14]. The Cog robotic platform uses a learned mapping between retinal displacement to motor command to compensate for motion sensed from optical flow and vestibular signals [3]. The Babyrobot platform [17] uses a neural-network stabilization algorithm incorporating both visual and inertial input [21]. The iCub robot [2] uses a chaotic controller that relies on visual feedback to stabilize the camera motion [8]. The camera stabilization system in [24] incorporates a controller based on the feedback error learning model of the cerebellum as proposed by Kawato [13]. A recent model of the cerebellum uses a recurrent architecture to solve the motor-error problem associated with feedback error learning [7, 6]. A stabilization system based on this recurrent model was implemented on a pneumatic-air-muscle driven eye in [14].

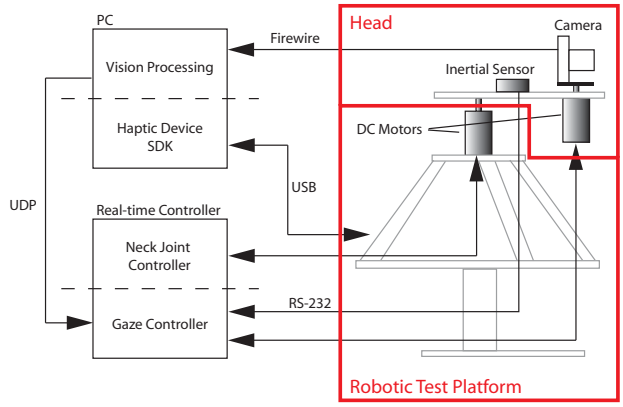


Figure 2: Overview of the hardware setup used to test the gaze control algorithm. The gaze controller uses only the visual, inertial, and camera position information to compensate the camera motion generated by the test platform.

Purely image-based stabilization and tracking has been studied for a long time under the name *visual servoing* [5]. Compared to systems using inertial sensors, visual servoing is better suited to tracking of slow phenomena, due to the limited sampling rate of most cameras. A notable exception is the 1000 frames per second camera system described in [19]. The system relies on a custom hardware setup consisting of 128 Field Programmable Gate Array chips to process an image with a resolution of 128x128 pixels. It would also require strong scene illumination or wide lens aperture to compensate for the high frame rate. Therefore this approach is not suitable for many applications.

Recently, there has been increased interest in developing high performance hardware for active vision. A notable example is reported in [25]. A head mounted camera system was stabilized using the eye's own Vestibulo-Ocular Reflex (VOR) and Optokinetic Reflex (OKR) by continuously tracking and reproducing the eye orientation.

Gaze stabilization is also provided recently by several companies for applications in UAVs and movie production. Examples include systems from Freefly, DJI, and Intuitive Aerial. However, the details of these systems are not known.

3. SYSTEM OVERVIEW

The system hardware consists of the following major components (see Figure 2): (1) Vision system; (2) Head, consisting of a camera, motor, and IMU; (3) Test platform, a high speed robot (3 DOF translational + 1 DOF rotational); and (4) Control computers.

3.1 Vision System

The vision system uses a Firewire camera running at 60 frames per second with a resolution of 648 × 488 (Dragonfly 2, Point Grey, Richmond, BC). The lens used provides a 74° horizontal viewing angle. The load inertia of the camera is minimized by only mounting the remote head on the motor while the processing board remains fixed to the head platform. The vision software described in Section 5 was developed using the OpenCV library.

3.2 Head

The camera is actuated using a geared (262:1) DC motor (MICROMO, Clearwater, FL) approximately about its optical center. The gearing gives the attached 16 lines per revolution magnetic encoder a resolution of 0.02°. This is much higher than the backlash due to the gearbox itself. The IMU (3DM-GX3-25, MicroStrain, Williston, VT) provides both translational acceleration and angular rate information. It has a full-scale range of ±5g and ±300°/sec with a 17-bit resolution. The inertial measurements are sent to the real-time controller through a serial (RS-232) connection at 1000Hz.

3.3 Test Platform

The head can be used while mounted on any mobile platform. However, to carefully test system performance we built a platform consisting of a high-speed translation stage which can move the head with 3 DOF, on which is mounted a rotation stage (called the "neck") that can rotate the head in the horizontal plane.

The translation stage consists of a commercial haptic device based on the delta parallel mechanism (Force Dimension, Nyon, Switzerland). Its parallel kinematic structure enables a high end-effector force (20 N continuous) and a large workspace (40 cm diameter). These properties are useful in moving the rest of the platform (including the head) quickly. The base of the delta mechanism is oriented such that it is parallel to the ground as shown in Figure 3 and controlled through USB by a PC. The neck is driven by a geared (14:1) DC Motor (MICROMO) with a 1024 lines per revolution optical encoder.

3.4 Control computers

The vision software and haptic device control execute on a generic PC. The rest of the setup is controlled by a hard real-time controller equipped with various I/O modules running at a sample rate of 0.5 msec (xPC Target, MathWorks Inc., Natick, MA). A 16-bit digital-to-analog board drives the linear motor amplifiers (LSC 30/2, Maxon motor AG, Sachseln, Switzerland) connected to the camera and neck joint DC motors.

4. ADAPTIVE CONTROLLER FOR GAZE STABILIZATION

Gaze stabilization requires the camera direction to be maintained with respect to the world or object of interest. While the camera direction can be measured using visual information, this is not always feasible when fast motion is involved due to its limited sampling rate. This is an important and subtle point: the sampling rate has to be limited due to the optical requirements of the camera, available illumination, and delays in visual processing. However, this results in motion blur in the images, making visual processing even more difficult. Instead, our gaze stabilization algorithm relies on the much faster IMU velocity data to drive a velocity compensation loop. We then add an outer position tracking loop, driven by vision information, to avoid drift.

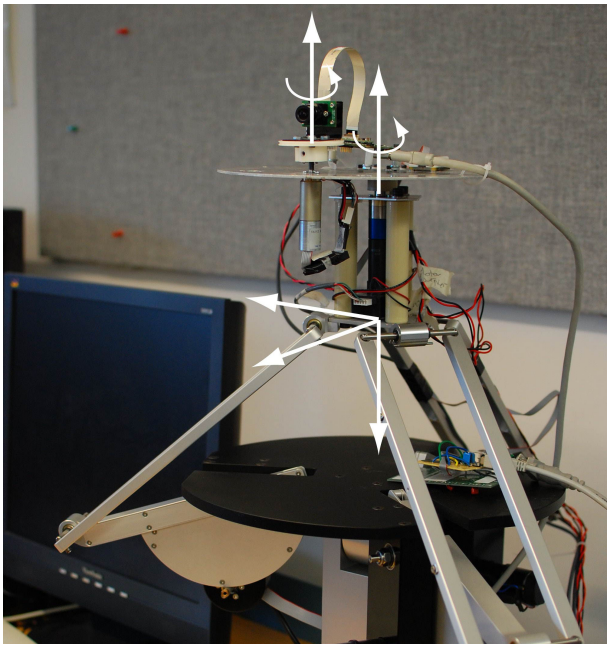


Figure 3: The head platform mounted on the robotic test platform with the axes of motion indicated. The camera head is actuated by a DC motor allowing a rotation about an axis perpendicular to the ground. The test platform rotational axis (at the neck joint) is parallel to the camera's and is colinear to one of its translational axis.

To perform velocity compensation, we implemented a model of the rotational VOR similar to the one described in [7] (see box in figure 4). The model's recurrent architecture allows for an adaptive controller which employs gradient descent learning without requiring a complicated inverse plant model. However, unlike [7], our brainstem is a simple gain, representing the simplest model of the plant inverse. It performs basic compensation to the velocity sensed by the vestibular system (inertial sensor). This simple plant inverse model is improved through the modulation of the *inputs* to the brainstem controller by the cerebellum. The cerebellum is modeled as an adaptive filter trained using a *distal* (sensory) signal (here the retinal slip).

Retinal slip. This term is somewhat ambiguously used in the literature, and needs some clarification before we start. We define the term to mean the angular velocity of the eye, as measured using image motion. This is a vector quantity in general, though in the planar case it is a scalar. It has units of radians (or degrees) per second. In a discrete setting, it is more useful to interpret this angular velocity in terms of the image motion between consecutive video frames. Therefore we will, on occasion, change the unit of angle to be the angle subtended by a standard pixel (typically at the center of the image), and unit of time to be the inter-frame interval, and express retinal slip as pixels/frame.

Notation. We use leading superscripts and subscripts to

denote coordinate frames: ${}^{w}_{h}\omega$ refers to the angular velocity ω of frame h with respect to frame w. We use w, h, l to denote world (inertial), head, and camera frame respectively.

The plant inverse modeled by the brainstem and cerebellum forms a feedforward controller to track an input angular velocity signal, ${}^{h}_{l}\omega_{\text{des}}$. To achieve disturbance cancellation in VOR, this signal should correspond to the negative of the head velocity measured by the vestibular system

$$ {}^{h}_{l}\omega_{\text{des}} = -{}^{w}_{h}\hat{\omega}. \tag{1} $$

The distal error signal is defined as

$$ e_c = {}^{h}_{l}\omega - {}^{h}_{l}\omega_{\text{des}}. \tag{2} $$

One of the advantages of VOR stabilization is that this error in the motion of the camera in the head can now be estimated from the *retinal slip* ${}^{w}_{l}\hat{\omega}$, i.e., the residual velocity of the eye relative to the visual world (see Section 5 for more information). This makes sense since, ignoring the non-collinearity of the rotational axes for the moment, the kinematic relationship between the measured variables for rotational VOR is given by

$$ {}^{w}_{l}\omega = {}^{w}_{h}\omega + {}^{h}_{l}\omega\,. \tag{3} $$

Substituting Eqs. 1 and 3 in Eq. 2 we have

$$ e_c = {}^{w}_{h}\hat{\omega} + {}^{h}_{l}\omega = {}^{w}_{l}\hat{\omega}. \tag{4} $$

This analysis could be extended to include camera translations (or equivalently, by Chasles' theorem, rotation about a different point). However, it is well known that errors due to camera rotation are much more significant than those due to translation (e.g., [27]), so we can safely ignore translations and small offsets of the rotation axis unless the visual targets are very close to the camera.

We implement the brainstem filter as the inverse of the motor speed constant of the DC motor driving the camera; this is a parameter that is easily obtained from the motor specifications. The cerebellum is implemented as a finite impulse response (FIR) filter with the following transfer function

$$ C(q) = \sum_{k=1}^{K} w_k q^{-kT_t}\,, \tag{5} $$

where q is the forward shift operator, T_t is the tap delay, and w_k are the adaptive weights that define the filter. We use $K = 160$ taps with a tap delay of $T_t = 10$ ms, resulting in a filter of length 1.6 sec.

Each time a retinal slip signal arrives, the weights are modified with the update rule

$$ \Delta w_k = -\gamma u(t - kT_t) e_c(t)\,, \tag{6} $$

where γ is the learning rate, appropriate values of γ were found to be in the range $\gamma \in [10^{-5}, 3 \times 10^{-5}]$. Lower values give very slow learning, and higher values result in an unstable filter due to the noise present in the retinal slip signal.

As the controller runs faster than the tap delay, learning is handled asynchronously. The weights are updated as soon as more than one tap delay has passed and a new retinal slip value is available.

Position tracking is performed similar to Shibata et al. [24], who add a proportional, or *P-controller* to convert the position error into a velocity error signal (e_p). This is added

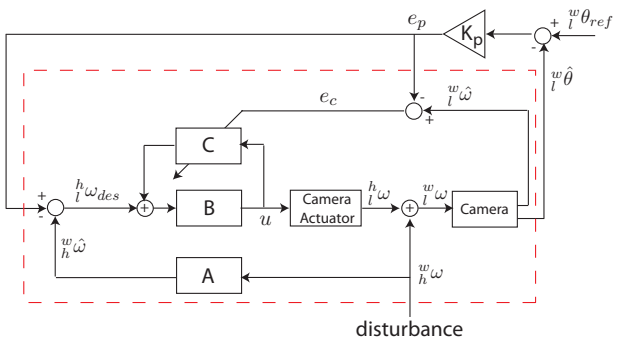


Figure 4: Our gaze stabilization architecture consisting of the velocity tracking inner loop (inside the dashed box) and a position tracking outer loop. The angular rate inertial sensor (A) provides an estimate of the rotational velocity of the head platform. In our Cerebellum-like recurrent design, the desired camera velocity is augmented with the output of the cerebellum (C) and used by the brainstem (B) to generate the plant control signal (u).

with the negative of the head velocity estimate to form the new signal tracked by the inner velocity loop. The distal error signal is also modified to reflect the change in desired velocity. The modified system architecture is illustrated in figure 4. We have in effect implemented a model of the OKR: the camera tracks the motion of the visual field even in the absence of head motion.

5. RETINAL SLIP ESTIMATION

Retinal slip is the main signal that facilitates learning of both VOR, fixation and OKR. Here we describe how it is estimated robustly.

5.1 Multi-resolution KLT

We make use of the multi-resolution KLT tracker [16, 23] as implemented in OpenCV. The KLT algorithm tracks small rectangular patches by finding the sub-pixel shift that aligns the Taylor expansion of each patch at two different time instances. The original KLT tracker is accurate, but can only handle small displacements, as it requires a good initial guess in order to work.

The OpenCV implementation makes use of coarse to fine search to also handle large displacements. Here alignment happens first at a coarse resolution. This first alignment is then used as an initial guess at the next finer resolution. This process is repeated until the finest resolution is reached.

5.2 Crosschecking and Resolution Adaptivity

As the KLT algorithm tracks small patches, it has problems near depth-discontinuities (where half the patch may look different in one of the frames) it also frequently matches the wrong regions if a repetitive texture is viewed. Most of these erroneous matches can be detected, and subsequently eliminated by adding a crosschecking step [1]. This step tracks each patch, first forwards and then backwards in time. Points that do not end up where they started (within a small tolerance) are rejected.

During fast rotational motions the KLT algorithm often

fails to find correspondences due to motion blur. For this reason we have added a layer on top of OpenCVs KLT. This layer detects frames where many regions fail to be tracked, and subsequently reduces the resolution of the images fed into KLT one octave at a time.

5.3 Estimation of 3D Camera Rotation

The coordinates of points tracked by KLT are expressed in the image grid, and thus depend on both the focal length and the optics of the camera. We convert the coordinates to normalised image coordinates using calibration parameters found from calibration using a planar target [28]. The projection model assumes that image coordinates $\mathbf{x}$ are generated from 3D points $\mathbf{X}$ as:

$$\mathbf{x} = f(\tilde{\mathbf{x}}, k_1, k_2) \quad \text{where} \quad \tilde{\mathbf{x}} = \mathbf{K}\mathbf{X}\,. \tag{7}$$

Here k_1 and k_2 are the lens distortion parameters, and $\mathbf{K}$ is the intrinsic camera matrix, all provided by the calibration.

Using the camera calibration, we find the normalised image coordinates $\mathbf{u}$, as:

$$\mathbf{u} = \mathbf{K}^{-1} f^{-1}(\mathbf{x}, k_1, k_2)\,. \tag{8}$$

The normalised coordinates are homogeneous 3-element vectors, and they have the desirable property that they are proportional to the 3D coordinates $\mathbf{X}$. By normalising them to unit length, we obtain the projection of the 3D points onto the unit sphere.

$$\hat{\mathbf{u}} = \mathbf{u}/\sqrt{u_1^2 + u_2^2 + u_3^2}\,. \tag{9}$$

Using projections of a set of points onto the unit sphere at two different time instances, we can compute the relative 3D rotation of the camera using the Orthogonal Procrustes algorithm [22]. Consider a set of normalised points $\mathbf{U} = [\hat{\mathbf{u}}_1 \ \ldots \ \hat{\mathbf{u}}_N]$ and the corresponding points at a different time instant $\mathbf{V} = [\hat{\mathbf{v}}_1 \ \ldots \ \hat{\mathbf{v}}_N]$. Assuming that the camera has undergone a pure rotation, they should be related as $\mathbf{U} = \mathbf{R}\mathbf{V}$. The Orthogonal Procrustes algorithm finds the least-squares approximation of the unknown rotation $\mathbf{R}$ by solving the following problem:

$$\arg\min_{\mathbf{R}} ||\mathbf{U} - \mathbf{R}\mathbf{V}||^2 \ , \text{ subject to } \mathbf{R}^T\mathbf{R} = \mathbf{I}\,. \tag{10}$$

Using the singular value decomposition(SVD) of the matrix $\mathbf{U}\mathbf{V}^T$ the solution becomes [9]:

$$\mathbf{R} = \mathbf{A}\mathbf{B}^T \text{ where } \mathbf{A}\mathbf{D}\mathbf{B}^T = \text{svd}(\mathbf{U}\mathbf{V}^T)\,. \tag{11}$$

This is how we find the 3D camera rotation $\mathbf{R}$ between two consecutive frames.

The rotation matrix $\mathbf{R}$ is related to the angular velocity vector $\boldsymbol{\omega} = [\omega_1\ \omega_2\ \omega_3]^T$ through the matrix exponent

$$\mathbf{R} = \exp\left(\begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} t\right), \tag{12}$$

where t is the inter-frame time interval. For rotation matrices the logarithm has a closed form expression, see e.g. [11], result A4.6. As our camera currently only has one axis of rotation, we compute the angular velocity about that axis by projection of the angular velocity vector onto the camera rotation axis, $\hat{\mathbf{n}}$,

$$\omega = \hat{\mathbf{n}}^T\boldsymbol{\omega}\,. \tag{13}$$

Note that this gives us the angular velocity in rad/frame.

6. PERFORMANCE EVALUATION

In this section and the next we evaluate the performance of the system. Please also see the supplementary video which shows the system in action.

6.1 Retinal Slip Accuracy

We checked the controller stabilization performance by first disturbing the system with a sinusoidal position signal of varying frequency. The frequency ranges from 0.16 to 0.64 Hz with a corresponding maximum speed from 57°/sec to 230°/sec respectively. The maximum velocity encountered in this test is close to the maximal speed of the camera motor. Figure 5a shows a snapshot of the disturbance velocity profile and the corresponding stabilization response from the camera. The retinal slip trace in the figure corresponds to the velocity of the features as seen by the camera. We believe that the retinal slip is the most important metric for evaluating the stabilization performance since the goal of a camera stabilization system is to minimize image movement. Note that the results are obtained after the cerebellar weights have been learned. We observe that the difference between disturbance velocity and camera velocity corresponds to the amount of retinal slip as expected. Most of the error occurs as the camera velocity lags behind during a change in direction.

Although velocity tracking such as shown in figure 5a is commonly used to evaluate VOR performance, an often overlooked aspect of VOR is position tracking. The position tracking performance of the system, corresponding to the same time period as figure 5a is shown in figure 5b. Here, the disturbance causes a maximum rotational displacement of ±56° while the stabilization system tracking error is bounded to 2°.

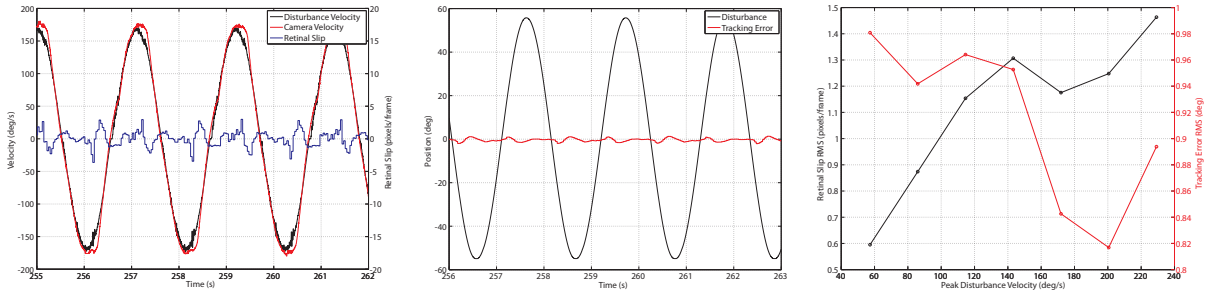
The system performance for different disturbance speeds is shown in figure 5c. As expected, the retinal slip tends to increase as the disturbance velocity increases. On the other hand the tracking error appears to be more robust to changes in the disturbance speed. We see that the stabilization system gives, for a disturbance up to 230°/sec, an RMS tracking error of less than 1° and a frame-to-frame image motion of less than 1.5 pixels. Figure 1 illustrates the motion blur effect caused by a disturbance of such speed and the large improvement obtained from using the gaze-stabilization system.

We also tested the stabilization algorithm with a colored noise disturbance. This is representative to the disturbance experienced by, for example, a vehicle travelling over a rough terrain. In this test we use white noise bandpass filtered from 0.5 to 1.5 Hz as a disturbance, similar to [14]. Figure 6 shows the response of the setup to the noise disturbance. After the cerebellar weights have been learned, the system has a retinal slip RMS of 2.1 pixels/frame.

We can clearly observe the effect of learning in the cerebellum by setting the gain in the brainstem path to be 50% of the correct gain. This causes the initial stabilization performance, before the cerebellum weights have been trained, to be very undergained. As the cerebellum becomes trained, the retinal slip quickly decreases. Figure 7 shows the speed at which the cerebellum learns and compensates for the deficient inverse plant model assumed by the brainstem.

6.2 User-driven Disturbance

Along with the dynamic platform, we have also tested



(a) Velocity tracking (b) Position tracking (c) RMS error of velocity and position tracking for different velocities

Figure 5: Tracking performance to a sinusoidal disturbance. (a) and (b) show tracking for a disturbance with a maximum velocity of 172°/sec. (c) shows the stabilization performance to different peak velocities caused by sinusoidal disturbance of various frequency. The sign of the camera velocity has been changed for easier visualization.

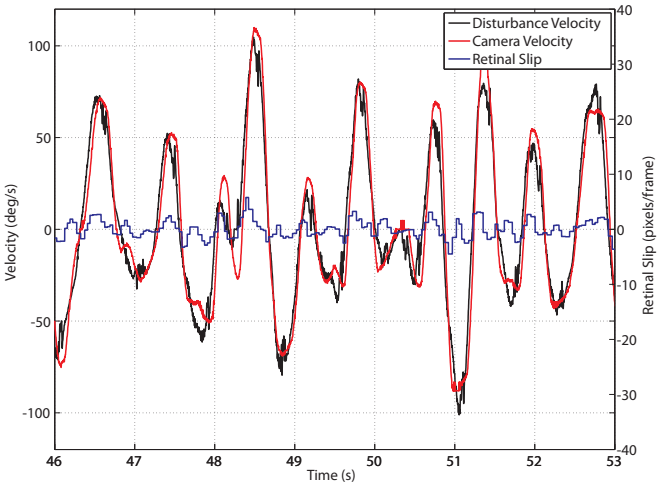


Figure 6: System performance to a bandpass noise disturbance. The sign of the camera velocity has been changed for easier visualization.

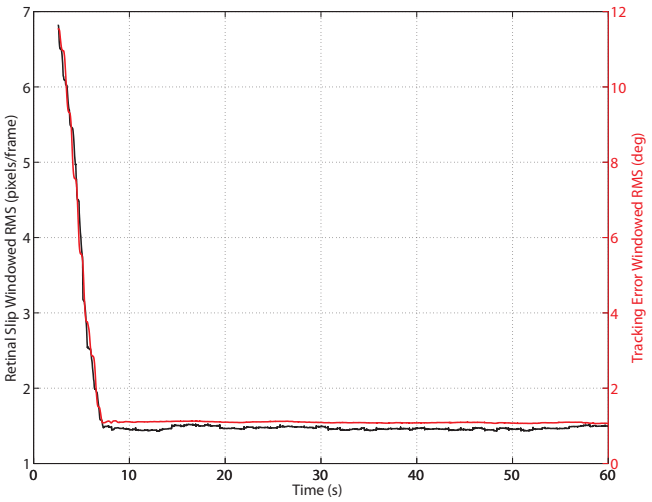


Figure 7: With the brainstem gain set to 50% of the correct value. Position and velocity tracking RMS error (with 5 seconds window) to a sinusoidal disturbance signal with a maximum velocity of 230°/sec

our gaze-stabilization system by mounting it on a wheeled-platform and disturbing it by hand as shown in figure 8. Even though this does not have the repeatability offered by the dynamic platform, it provides a clear delineation between the disturbance source and the stabilization algorithm. Figure 9 shows a sample user-driven disturbance (as measured by the IMU) and the corresponding stabilization result.

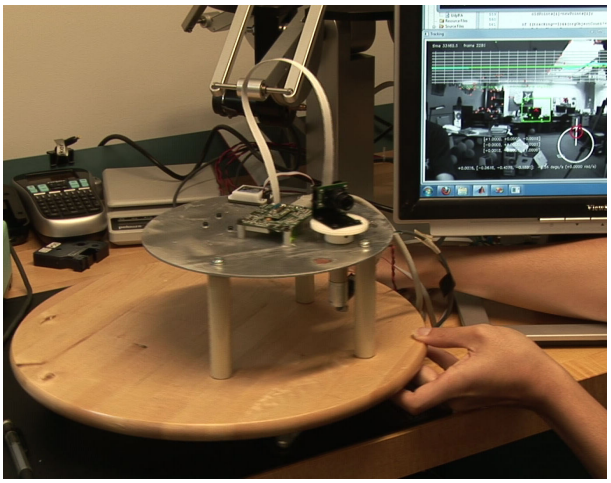


Figure 8: The wheeled-platform used to subject our gaze-stabilization system with user-driven disturbance.

7. EFFECT OF FAST MOVEMENTS ON COMPUTER VISION ALGORITHMS

Common computer vision tasks, such as detection and recognition work better with sharp images, and their use on mobile platforms should thus benefit from active gaze stabilization. In this Section we demonstrate this with the very common task of face detection. The OpenCV implementation of a Haar Feature-based cascade classifier, first proposed by Viola and Jones [26], is used as the face detector.

We tested the effect of fast movements on the face detec-

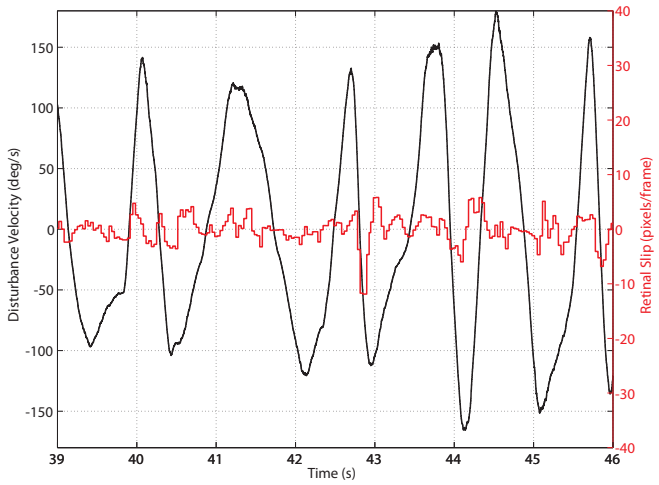


Figure 9: System performance to a user-driven disturbance. The disturbance velocity is obtained directly from the angular rate measurement of the IMU.

tion algorithm by placing a set of printed faces within the camera field of view at a fixed distance from the camera. The printed faces were used as substitute for human subjects as they can remain fixed in the scene across the different experiments for fair comparsion. We then applied sinusoidal disturbances with increasing velocity and capture the images recorded by the camera. This is then repeated with the faces placed at a different distance. The captured frames in Figure 10 show that aside from the targets disappearing from the camera field of view, significant motion blurring also occurs at higher speeds. Both of these effects negatively affect the face detection algorithm. Even at 46°/sec, the face detection algorithm already misses some of the faces, at 230°/sec none of the faces are detected. Figure 11 shows the results for the frames which have the same field of view as the undisturbed case. We see that the face detection performance deteriorates as the speed of the disturbance, and consequently motion blur, increases.

We repeated the experiment with the stabilization system turned on. The stabilization causes the field of view to be maintained under fast movements and also minimizes the blurring as shown in Figure 10c. The face detection algorithm performs consistently well even under increasing disturbance speed (see Figure 11).

8. CONCLUSIONS

Our adaptive algorithm is composed of an inner velocity loop driven by the fast IMU velocity data and an outer position tracking loop which provides drift correction. The velocity loop mimics the architecture of the biological VOR system whereby inertial information provides velocity compensation and the system is continuously tuned using the retinal slip error signal. The outer position loop uses position information from the camera to track an object, giving a behavior similar to the OKR. By adding a fast velocity loop instead of just using the position loop, we can incorporate inertial information to compensate for fast platform motion. This keeps our setup simple and affordable as inertial measurements can be obtained at high sample rate much more easily than vision data at the same rate.

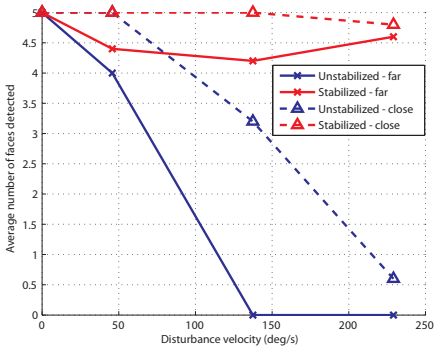


Figure 11: Relationship between disturbance velocity and face detection performance when all 5 printed faces are within field of view. Each data point corresponds to an average of over 5 samples, ignoring false positives.

We evaluated the performance of the stabilization algorithm with various types of input, including sinusoidal head movements, colored noise head movements, and random user inputs. We also demonstrated how active gaze stabilization allows face recognition to work in circumstances where it would otherwise fail completely. The reason for this is that visual accuity can be sustained. Humanoids and other interactive robots should directly benefit from this, as they make heavy use of face detection. In general, loss of visual acuity is a real and important issue encountered in all robotic systems with moving cameras. Similar performance improvements are to be expected for any vision algorithm that requires good visual acuity.

Acknowledgments

This work was funded in part by grants from NSERC, CFI, PWIAS, Canada Research Chairs Program, and the Swedish research foundation grant for the project *Learnable Camera Motion Models*, and by Linköping University.

9. REFERENCES

[1] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Proceedings 2007 IEEE International Conference on Computer Vision (ICCV)*, 2007.

[2] R. Beira, M. Lopes, M. Praça, J. Santos-Victor, A. Bernardino, G. Metta, F. Becchi, and R. Saltarén. Design of the robot-cub (icub) head. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 94 –100, Orlando, Florida, May 2006.

[3] R. A. Brooks, C. Breazeal, M. Marjanović, B. Scassellati, and M. M. Williamson. Computation for metaphors, analogy, and agents. chapter The cog project: building a humanoid robot, pages 52–87. Springer-Verlag, Berlin, Heidelberg, 1999.

[4] Canon. What is optical image stabilizer? `http://www.canon.com/bctv/faq/optis.html`, December 2010.

[5] P. I. Corke. *High-performance visual closed-loop robot control*. PhD thesis, Mechanical and Manufacturing Engineering, University of Melbourne, 1994.

[6] P. Dean and J. Porrill. Oculomotor anatomy and the

(a) Unstabilized, peak velocity 46°/sec (b) Unstabilized, peak velocity 230°/sec (c) Our stabilization method, peak velocity 230°/sec

Figure 10: Image frames with faces, captured by the camera (60 fps) with and without stabilization. Detected faces are marked with a red box. Our system performs well even at 230°/sec.

motor-error problem: the role of the paramedian tract nuclei. *Progress in Brain Research*, 171:177–186, 2008.

[7] P. Dean, J. Porrill, and J. Stone. Decorrelation control by the cerebellum achieves oculomotor plant compensation in simulated vestibulo-ocular reflex. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 269(1503):1895–1904, 2002.

[8] B. Durán, Y. Kuniyoshi, and G. Sandini. Eyes-neck coordination using chaos. In H. Bruyninckx, L. Preucil, and M. Kulich, editors, *European Robotics Symposium 2008*, volume 44 of *Springer Tracts in Advanced Robotics*, pages 83–92. Springer Berlin / Heidelberg, 2008.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1983.

[10] W. Gunthner, P. Wagner, and H. Ulbrich. An Inertially Stabilised Vehicle Camera System-Hardware, Algorithms, Test Drives. In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pages 3815–3820. IEEE, 2007.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, 2nd ed.* Cambridge University Press, New York, NY, USA, 2003.

[12] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29:30:1–30:9, July 2010.

[13] M. Kawato. Feedback-error-learning neural network for supervised motor learning. *Advanced neural computers*, 6(3):365–372, 1990.

[14] A. Lenz, S. R. Anderson, A. Pipe, C. Melhuish, P. Dean, and J. Porrill. Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic arificial muscles. *IEEE Transactions on Systems, man and cybernetics*, 39(6):1420–1433, December 2009.

[15] Y. Liang, H. Tyan, S. Chang, H. Liao, and S. Chen. Video stabilization for a camcorder mounted on a moving vehicle. *Vehicular Technology, IEEE Transactions on*, 53(6):1636–1648, 2004.

[16] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[17] G. Metta. *Babyrobot: A study into sensori-motor development*. PhD thesis, University of Genoa, LIRA-Lab, DIST, Genoa, Italy, 2000.

[18] C. Morimoto and R. Chellappa. Fast electronic digital image stabilization for off-road navigation. *Real-Time Imaging*, 2(5):285–296, 1996.

[19] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno. 1ms column parallel vision system and it's application of high speed target tracking. In *Proceedings 2000 IEEE International Conference on Robotics and Automation (ICRA)*, pages 650–655, San Francisco, CA, April 2000.

[20] Nikon. Nikon vr. `http://imaging.nikon.com/products/imaging/technology/vr/`, December 2010.

[21] F. Panerai, G. Metta, and G. Sandini. Learning visual stabilization reflexes in robots with moving eyes. *Neurocomputing*, 48:323–337, 2002.

[22] P. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966. 10.1007/BF02289451.

[23] J. Shi and C. Tomasi. Good features to track. In *Proceedings 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, 1994.

[24] T. Shibata, S. Vijayakumar, J. Conradt, and S. Schaal. Biomimetic oculomotor control. *Adaptive Behaviour*, 9(3-4):189–207, 2001.

[25] T. Villgrattner and H. Ulbrich. Design and control of a compact high-dynamic camera-orientation system. *Mechatronics, IEEE/ASME Transactions on*, 16(2):221–231, 2011.

[26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages I–511 – I–518 vol.1, Kauai, Hawaii, 2001.

[27] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, USA, June 2010. IEEE Computer Society, IEEE.

[28] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.