2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

A Real-Time Inertial Motion Blur Metric: Application to Frame Triggering Based Motion Blur Minimization*

Mehmet Mutlu[1], Afsar Saranli[2] and Uluc Saranli[3]

***Abstract*— Mobile robots suffer from sensory data corruption due to body oscillations and disturbances. In particular, information loss on images captured with onboard cameras can be very high, and such loss may become irreversible or computationally costly to undo. In this paper, we propose a novel method to minimize average motion blur captured by such mobile visual sensors. To this end, we derive a motion blur metric (MMBM) that can be computed in real-time by using only inertial sensor measurements and validate it through comparisons with optic flow computations. The applicability of MMBM is illustrated through a motion blur minimizing system implemented on the SensoRHex hexapod robot by externally triggering an onboard camera based on MMBM values computed in real-time while the robot is walking straight on a flat surface. The resulting motion blur is compared to motion blur levels obtained with a regular, fixed frame-rate image acquisition schedule by both qualitative inspection and using a blind blur metric on captured images. MMBM based motion blur minimization system not only reduces average motion blur, but also avoids frames with extreme motion blur before an image gets corrupted by appropriately delaying the triggering of frame acquisition.**

I. INTRODUCTION

All mobile robots, with legged morphologies in particular, exhibit unpredictable body oscillations due to their own structure and disturbances from their environment. For dynamically dextrous legged robots such as RHex platform instances [1], these body oscillations result from the robot's own locomotory behaviors and are hence unavoidable.

These undesirable motion disturbances can degrade the performance of sensors mounted on the robot. The performance of spatial optic sensors such as cameras are particularly susceptible to ego motion, with angular disturbances having particularly significant effects on the quality of captured frames. The most significant distortion for metrological images is the motion blur. Even though motion blur itself can be used for useful tasks such as computing the motion, velocity and orientation of a camera or objects [2], [3], or identifying whether an image is manipulated [4], it is usually undesirable for applications requiring precise features to be extracted from images. It is known that motion blur negatively affects many vision and image processing algorithms, particularly those requiring feature extraction and tracking [5]. In [6], for example, a bipedal robot is occasinally forced to stop so that frames without motion blur can be obtained and features can be precisely located.

Fortunately, motion disturbances within certain applications may exhibit properties that can be exploited. For example, dynamic legged robots performing stable locomotion on flat surfaces exhibit quasi-periodic body oscillations arising from limit cycles associated with their behavioral primitives. In fact, many land-based mobile robots are likely to exhibit such quasi-periodic trajectories in cross-sections of their state space such as their body orientation and angular velocities.

In this paper, our goal is to reduce motion blur from images captured by a camera mounted on a dynamic robotic base by using a system to compute expected motion blur prior to capturing a frame. We use angular velocity measurements to predict average motion blur, and attempt to minimize its corruptive effects thereby increasing the performance of computer vision algorithms that require feature extraction. An important criteria is that such detection and measurement pre-processing should be computationally inexpensive to be implemented in real-time with minimal delay.

Clearly, the exact amount of motion blur on the camera image plane depends on the camera motion during the exposure period. Even though external object motion also contributes to motion blur, our focus in this paper is on dominant ego motion that corrupts the entire frame with motion blur. Our hypothesis is that the predictability of quasi-periodic body oscillations of a legged robot can be exploited to avoid exposure periods where excessive motion blur is expected to corrupt the image. Even by incorporating a simple avoidance strategy on the timings of frame capture, motion blur can be reduced on the average and excessive motion blur can be avoided. On favorable surfaces where body oscillations become more predictable, the benefits can be increased by signal prediction approaches.

The first step in applying such a technique for improving motion blur performance for a video stream is to have a motion blur metric that can be computed in real-time. Consequently, a primary contribution of our paper is the derivation of an average motion blur metric, which we call the Motion based Motion Blur Metric (MMBM), based on inertial motion measurements obtained through a gyro.

The paper is organized as follows: Section II begins by presenting relevant literature for our study, followed by Section III where our new motion blur metric based on angular velocity measurements, MMBM, is presented. Section IV provides the validation of the MMBM, followed by Section

*This work is supported by TUBITAK under the project code 110E120

[1]Mehmet Mutlu is with the Department of Electrical and Electronics Engineering, Middle East Technical University, 06800 Ankara, Turkey mehmet.mutlu@ieee.org

[2]Afsar Saranli is with the Department of Electrical and Electronics Engineering, Middle East Technical University, 06800 Ankara, Turkey afsars@metu.edu.tr

[3]Uluc Saranli is with the Computer Engineering Department, Middle East Technical University, 06800 Ankara, Turkey saranli@ceng.metu.edu.tr

V where MMBM is further approximated to satisfy real-time considerations. Finally, an experimental application of MMBM is presented in Section VI on our the SensoRHex robot platform to minimize motion blur during walking. Our experimental results are presented in Section VII.

II. Background on Motion Blur Reduction/Removal

There are both hardware and software methods to compensate for motion blur. Hardware methods are widely used by camera manufacturers. High end commercial cameras use lens or sensor motion techniques for image stabilization. Another commonly used hardware solution is mounting the camera on top of a stabilization platform. Gimballed platforms, Stewart platforms and their variants are commonly used for line-of-sight stabilization [7], [8]. Such platforms are complex, costly and often require complex control algorithms. Moreover, robust pose estimation is a challenging problem for highly dynamic robots due to inertial measurement drift [9]. It may be difficult to use Stabilization platforms on small scale robotic platforms due to their size, weight and effects on robot dynamics. Moving the camera can also be used to modify Power Spectral Function to increase performance of motion deblurring with deconvolution [10].

Software based motion blur removal techniques primarily focus on individual frames only after an image is captured with motion blur [11]. The downside of single frame software methods is that the deconvolution operation is ill-defined since some information is permanently lost due to the nature of motion blur. Deconvolution techniques are usually computationally costly as well and may be difficult to implement in real-time applications [12].

Inertial and visual sensors act as competent pairs [13], with inertial sensors used to obtain extra information on motion. The Point Spread Function (PSF), whose knowledge is necessary for deblurring, can be estimated with inertial measurements [14] or with a complementary camera [15] and deconvolution is applied for deblurring as a better alternative to blind deconvolution.

Camera shutter control is also used in certain applications to compensate for motion blur. The simplest solution is limiting exposure time, but the image Signal to Noise Ratio decreases due to the reduced amount of light integration. Special lighting is usually required for this approach to be successful. More complex methods also exist. Light integration pattern can be manipulated to minimize deconvolution noise [16]. Fusion of frames exposed at different durations on the same scene can also be used to reduce motion blur [17]. However, the computational complexity is a considerable burden of these approaches especially when they run on low-power onboard CPUs of a mobile robot.

In the present study, we focus on deriving a novel real-time metric using three-axis gyroscope measurements to predict motion blur that would result from the rotational motion of the camera. This metric is then used in a simple approach to reduce average motion blur while capturing a video sequence.

III. Motion based Motion Blur Metric (MMBM) Derivation

Motion blur caused by rotational camera motion can be derived when this motion is known. Exact formulation of motion blur requires the integration of light falling upon an image pixel over the frame exposure time [18]. Fig. 1 illustrates the timing of two successively captured frames whose exposure times can be identified as t_{e1} and t_{e2}. The aim of MMBM is to approximate camera rotation based average motion blur that will result from exposure during t_{e1} by using inertial measurements collected at the time instant t_{i1}. Deriving the exact motion blur with MMBM is impossible since it requires knowledge of future. Unless absolute camera motion is controlled or future camera motion can be precisely estimated with a motion model, this is impossible. However, MMBM can still give reasonable results in many applications since computation and trigger delay t_{d1} and exposure time t_{e1} can be sufficiently small that robot dynamics would restrict radical changes in rotational velocity. Furthermore, many causes of impulsive rotational velocity changes, such as leg touchdown instants of a legged robot, can be avoided by modeling them as a function of input and robot states and predicting their occurrences.

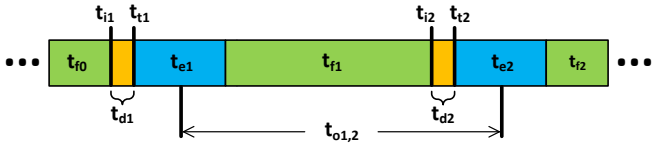


Fig. 1: Timing diagram of MMBM based camera triggering.

MMBM uses 3D rotational velocities of the camera measured with a gyro and approximately predicts optic flow at time instant t_{i1}. Since the effect of rotational motion on motion blur is more pronounced compared to that of translational motion for reasonable scene distances, MMBM is expected to be widely applicable.

A. Notation

Fig. 2 illustrates world (X,Y,Z) and image (U,V) coordinate frames; a point on the real world (x,y,z) and its projection on image plane (u,v); as well as the rotational velocities w_x, w_y and w_z of the camera.

B. Camera Model

A basic pinhole camera model is used for modeling the camera. In vector form, this can be written as,

$$\begin{bmatrix} u \\ v \end{bmatrix} = f \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \end{bmatrix}. \tag{1}$$

The associated inverse camera model is given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} z\frac{u}{f} \\ z\frac{v}{f} \\ z \end{bmatrix}. \tag{2}$$

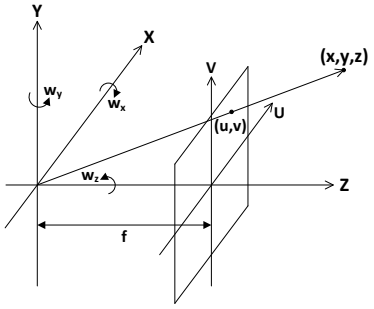


Fig. 2: Illustration of frames and definitions used in the derivation of the MMBM.

Our formulation assumes that the camera is pre-calibrated and leaves out intrinsic camera parameters such as distortion to keep the MMBM derivation as simple as possible.

C. Velocity Relation between World and Image Coordinates

The relation between a real world point and its corresponding projection on the image plane before and after rotational (R) and translational (T) camera motion is detailed in [19]. A spatially varying motion blur model that uses time-varying R and T obtained from inertial measurements during exposure time is proposed by [15]. Instead of using R, MMBM makes use of a world point's projected velocity on image sensor by using camera rotational velocities. The time derivative of the camera model is used to yield

$$\frac{d}{dt}\begin{bmatrix}u\\v\end{bmatrix} = f\begin{bmatrix}\frac{\dot{x}z-x\dot{z}}{z^2}\\ \frac{\dot{y}z-y\dot{z}}{z^2}\end{bmatrix} = \begin{bmatrix}\frac{f}{z} & 0 & -\frac{fx}{z^2}\\ 0 & \frac{f}{z} & -\frac{fy}{z^2}\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}. \quad (3)$$

A fixed point in world coordinates with the camera rotating on the origin is analogous to the point itself rotating around a fixed camera in the same frame of reference. The velocity of a point when rotated with respect to an arbitrary vector, w, passing through the origin of the point's frame of reference is defined as

$$\dot{P} = W \times P \quad (4)$$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}0 & -w_z & w_y\\ w_z & 0 & -w_x\\ -w_y & w_x & 0\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}. \quad (5)$$

D. The Definition of the MMBM

Obtaining the averaged optic flow requires the integration of instantaneous image plane optic flow vector magnitudes caused by camera rotation at time instant t_{i1}. This leads to the definition of our MMBM as

$$\mu := \frac{1}{\Delta u \Delta v}\int_{u_{min},v_{min}}^{u_{max},v_{max}} \sqrt{\dot{u}^2+\dot{v}^2}\,\mathrm{d}u,v \quad (6)$$

where Δu and Δv are defined as $(u_{max}-u_{min})$ and $(v_{max}-v_{min})$ respectively.

Inserting (3), (5) and (2) respectively into instantaneous optic flow vector in (6) to be able to explicitly evaluate the integral, we have

$$\begin{bmatrix}\dot{u}\\ \dot{v}\end{bmatrix} = \begin{bmatrix}\frac{f}{z} & 0 & -\frac{u}{z}\\ 0 & \frac{f}{z} & -\frac{v}{z}\end{bmatrix}\begin{bmatrix}0 & -w_z & w_y\\ w_z & 0 & -w_x\\ -w_y & w_x & 0\end{bmatrix}\begin{bmatrix}z\frac{u}{f}\\ z\frac{v}{f}\\ z\end{bmatrix}. \quad (7)$$

The expression of $(\dot{u}^2+\dot{v}^2)$ then becomes

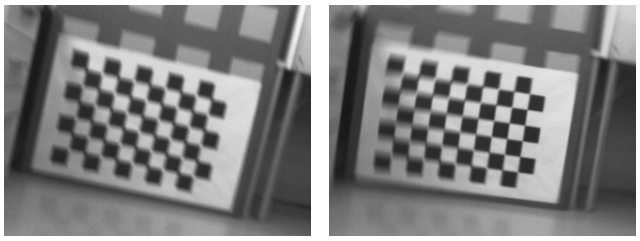
$$\begin{aligned}\dot{u}^2+\dot{v}^2 =& \frac{w_y^2}{f^2}\boldsymbol{u^4} + \frac{w_x^2}{f^2}\boldsymbol{v^4} + \frac{w_x^2+w_y^2}{f^2}\boldsymbol{u^2v^2}\\ &- 2\frac{w_xw_y}{f^2}\boldsymbol{u^3v} - 2\frac{w_xw_y}{f^2}\boldsymbol{uv^3}\\ &+ (2w_y^2+w_z^2)\boldsymbol{u^2} + (2w_x^2+w_z^2)\boldsymbol{v^2}\\ &- 4w_xw_y\boldsymbol{uv} - 2w_xw_zf\boldsymbol{u} - 2w_yw_zf\boldsymbol{v}\\ &+ (w_x^2+w_y^2)f^2.\end{aligned} \quad (8)$$

This expression can be used to compute (6) either numerically, or through the approximation proposed in Section V.

IV. Validation of the MMBM

MMBM calculates the derivative of optic flow at t_{i1}. The assumption of constant velocity motion from t_{i1} to the end of t_{e1} is required for the accuracy of MMBM. Under the assumption of constant camera rotational velocity, MMBM is proportional to the average of optic flow vector magnitudes.

In order to validate our metric, a hardware setup consisting of a camera, a 3D fiber-optic gyro and a PC was built. Camera and gyro axes were matched with a fixture, but no calibration was done to obtain further alignment information. Using handheld hardware, time synchronized image frames and gyro data samples were collected. The method proposed by [20] was used to find the optic flow between all successive frames on the collected images. Two of the successive images from the data set are shown in Fig. 3.



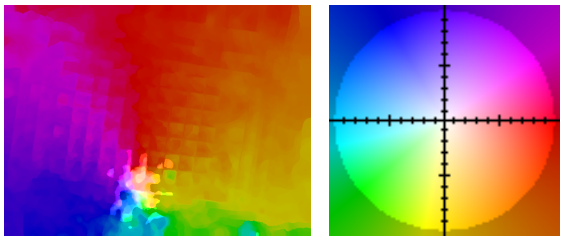
(a) (b)

Fig. 3: Two successive images from the sequence recorded to validate the MMBM.

For example, color coded optic flow field between Fig. 3(a) and Fig. 3(b) is shown in Fig. 4(b). Direction and intensity of optic flow vectors can be visualized with the help of the color map in Fig. 4(a).

We first compute the average magnitude of optic flow (AMOF) for each consecutive image frame. AMOFs are then compared with MMBM values calculated from gyro data only. Fig. 1 can be used to be clear on the timing of comparison. Optic flow algorithms assume that the exposure times of frames (t_{e1}&t_{e2}) are infinitesimal. Flow vectors give velocities of world points projected on to the image plane

from the beginning to the end of $t_{o1,2}$. Therefore, MMBM calculated with gyro data taken in the mid point of $t_{o1,2}$ is scaled and compared to AMOF. The relative scales of MMBM and AMOF are actually different since they are two different quantities. The main objective of validation is comparing the time behavior two waveforms since both give an idea about the magnitude of motion blur under aforementioned assumptions.



(a) (b)

Fig. 4: Optic flow field from images in Fig. 3: (a) The actual optic flow field, (b) optic flow color and direction map.

Comparison of MMBM and AMOF with the first data set is illustrated in Fig. 5. The first data set was collected with the camera rotating around the yaw axis and rotations on other axes negligibly small. Note that data sets also include a small amount of translation movement since they were collected with handheld hardware. Since the target is at a reasonable distance, the projection of translational movement onto the image plane was assumed to be negligible. Our results show that MMBM and AMOF waveforms have closely matching behavior. Extremum points on both waveforms are very close to each other. Note, also, that the sampling rate of the gyro (approx. 600Hz) and the camera (approx. 12fps) are very different. This is the reason why the MMBM plot seems to be continuous and AMOF is not. Red dots on the AMOF plot correspond to mid points of time durations $t_{o1,2}$ and AMOF data can only be calculated on the red dots, with the rest being a piecewise linear interpolation in between.

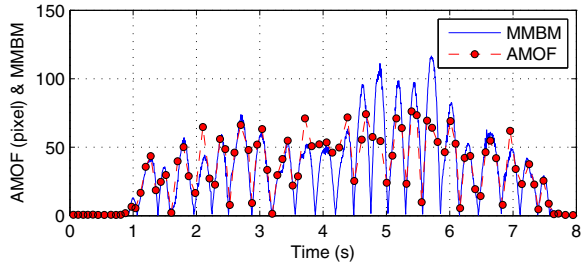


Fig. 5: Comparison of scaled MMBM and AMOF during yaw axis camera rotations.

On the second data set, the camera was subjected to a complex rotational motion with rotation with respect to each axis changed continuously and arbitrarily in time. It can be observed that discrepancies in Fig. 6 are more pronounced. The main reason for this is the slight increase in the frequency of rotations. Camera data starts to suffer from low sampling rate and the resulting aliasing. Moreover, Optic flow algorithms are affected from motion blur and hence AMOF deteriorates as a result. It is also useful to remind that, the current comparison is done with the assumption of constant velocity rotations. The second data set is hence at the limit of validity for comparing MMBM and AMOF. However, the general form of the waveforms are still consistent with each other.

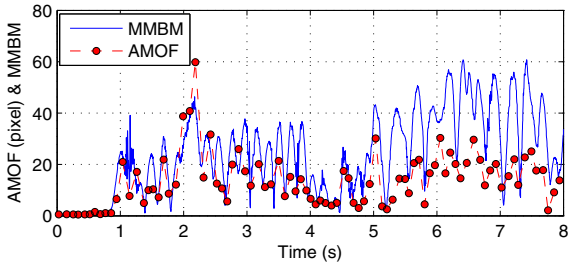


Fig. 6: Comparison of scaled MMBM and AMOF during arbitrary camera rotations.

V. Real Time Evaluation of MMBM

MMBM was evaluated numerically in Section IV since the analytic evaluation of the integral was not possible. Numeric integration is usually computationally costly and cannot be implemented in real-time applications. Consequently, MMBM was approximated with a Riemann sum to meet real time requirements.

Instead of evaluating the MMBM definition of (6) over the whole image plane, it was approximated with Riemann sum using square areas whose values are evaluated only at middle points shown in Fig. 7. The final MMBM calculation hence reduces to

$$\mu^* = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\dot{u_i}^2 + \dot{v_i}^2}dA. \qquad (9)$$

where $\dot{u}^2 + \dot{v}^2$ is given by Eq. (8) and dA is the region whose value is approximated with the exact value of a single, mid point. All summation regions are square and uniformly sampled from the image plane. Note that multiplication of dA in (9) can also be ignored since all regions have the same area and only the waveform of the MMBM is important. This approximation is reasonable since all image plane motion is the result of a single camera ego motion and therefore exhibit significant spatial smoothness.

Numerical evaluation and Riemann sum approximation of MMBM gives almost the same function form except small integration errors. The percentage error between numerical evaluation and Riemann sum approximation, $100 * (\mu - \mu^*)/\mu$, is shown in Fig. 8, with the small percentage errors justifying our choice of the number of samples used for the approximation.

VI. Application of MMBM: External Camera Shutter Triggering to Minimize Motion Blur

Our motion blur prediction and minimization system was implemented on the SensoRHex hexapod. The system con-

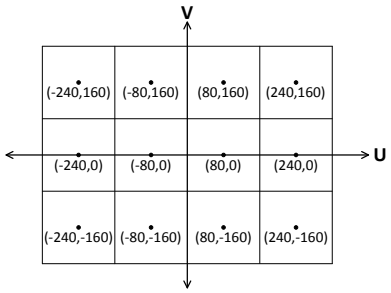


Fig. 7: Areas (dA) and value evaluation points used for Riemann sum approximation.

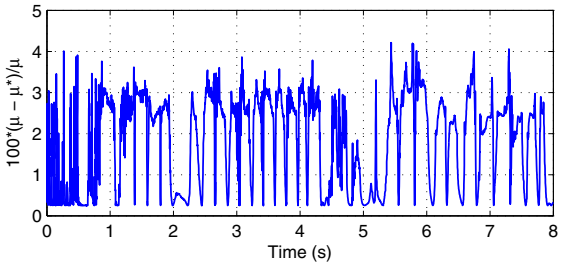


Fig. 8: Percentage error between the numeric evaluation and the Riemann sum approximation of MMBM (μ).

sists of a Fizoptika optic gyro, a PointGrey camera, a PIC based microcontroller board and a 500MHz Pentium class PC-104 CPU. Connections between hardware components are illustrated in Fig. 9.

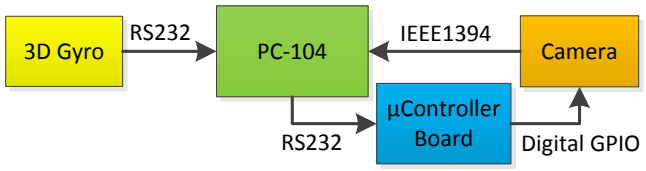


Fig. 9: Hardware structure of motion blur reduction system implemented on the SensoRHex hexapod.

All hardware components are located on SensoRHex. The gyro gives rotational velocities in each axis at 600Hz. MMBM is calculated for each gyro reading by a PC-104 CPU unit in real-time. Fig. 10 illustrates MMBM calculated during the normal walking mode of SensoRHex. The robot walks on a flat concrete surface. High MMBM values correspond to time instances where motion blur would be high if an image acquisition was triggered at that moment. Fig. 10 spans a duration slightly longer than two steps of SensoRHex and three leg touch down instances can be observed as the highest MMBM values on the plot. The quasi periodicity of body oscillations can also be observed. Exploiting such quasi periodicity, the camera can be triggered only when MMBM is below a certain threshold.

The camera was configured to work in its external triggering mode with the exposure time fixed to 70ms. Trigger signals were given by the PC and directed to the microcontroller board over a serial port. The microcontroller then generated

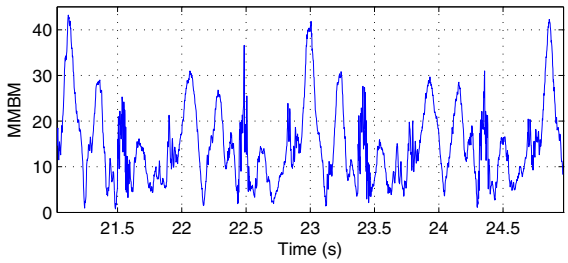
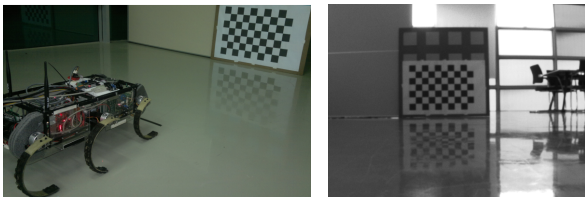


Fig. 10: MMBM while SensoRHex is walking straight on a flat concrete surface.

signals to trigger the camera. Finally, captured frames were transferred to the PC over the IEEE1394 interface.

VII. Experimental Results

SensoRHex was equipped with our proposed motion blur minimization setup. Two data sets were collected while SensoRHex was walking straight on a flat surface and pointing to a checkerboard pattern as shown in Fig. 11. The first data set incorporates frames captured at a fixed rate of 5fps and the second one uses our motion blur minimization system to determine capture timings. The camera was setup to have the same exposure time, 70ms, for both. Our motion blur minimization system waits at least 200ms between two successive frames to ensure a fair comparison with the 5fps fixed frame rate set. Once 200ms passes, the PC starts to calculate MMBM for each gyro reading. The trigger command is applied only if MMBM is below certain threshold.



(a) (b)

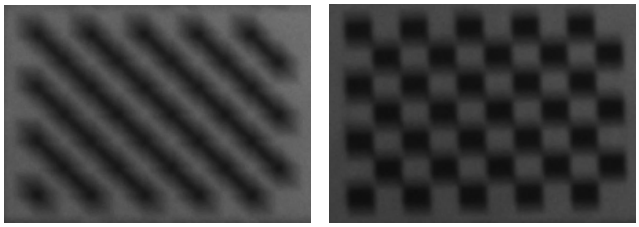
Fig. 11: Motion blur minimizing system implementation on SensoRHex. (a) Experiment area, (b) robot's point of view.

TABLE I: JNBM averages over frames captured during straight walk of SensoRHex on flat surface

Image Capture Method	JNBM average
Fixed Frame Rate @5fps	0.3564
MMBM Based Motion Blur Minimized Capture	0.4908

In order to quantitatively evaluate improvements on motion blur, checkerboard patterns were cropped from each frame and cropped images were evaluated with Just Noticeable Blur Metric (JNBM) [21]. However, blur amount detection within a single frame is an ill-defined problem and isolated JNBM results may not totally agree with human visual inspection, JNBM gives consistent results on the

average. JNBM for both data sets are presented in Table I. JNBM gives higher results for sharper images. Subjective inspection of frames also agree with the average JNBM results. In particular the most blurred frames are clearly avoided with our proposed system. Fig. 12(a) and Fig. 12(b) show hand-picked extremely blurred frame examples from both data sets.



(a) (b)

Fig. 12: Examples of hand-picked excessively blurred frames from SensoRHex walking: (a) Image from fixed frame rate capture (b) Image from external triggering with MMBM.

VIII. CONCLUSIONS

In this work, a new metric, MMBM, to evaluate camera rotation based average motion blur of an image was presented. MMBM is evaluated by using only gyro data in real-time. Finally, MMBM was used to minimize motion blur of captured images from an onboard camera mounted on the SensoRHex hexapod robot by externally triggering the camera only when MMBM dips below a certain threshold. Average motion blur of captured images is hence decreased and, more importantly, extremely blurred images are avoided. As a consequence, computer vision algorithms such as localization can run more efficiently.

In the present paper, a camera was externally triggered, but, a software trigger such as polling a image frame over data transmission line could have been applied if external triggering was not available. Another approach could have been using a fixed frame rate image acquisition and deciding whether or not to use an image after capturing it and before applying any computer vision algorithms on it.

MMBM can be used in many robotics applications that suffer from motion blur, especially, legged robots in particular. Quasi-periodicity of a legged robot's body oscillations can be exploited to pick up the best instants to capture a frame. Further improvements can be achieved by incorporating the motion model of a robot. Unfortunately, SensoRHex currently does not have any such models, but, optimal shutter triggering can be obtained with robots having a motion model. Even though a robot does not have any motion model, motion blur detection performance can be further increased by integrating signal prediction methods on MMBM.

Significant amount of motion blur has been avoided before it corrupts an image by using our method. Rest of the motion blur can be deblurred using methods available in the literature. For instance gyro data used to calculate MMBM can also be used to construct PSF for deconvolution based motion blur removal.

ACKNOWLEDGMENT

This work was partially supported by TUBITAK project 110E120.

REFERENCES

[1] U. Saranli, M. Buehler, and D. E. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.

[2] S. Dai and Y. Wu, "Motion from blur," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.

[3] H.-Y. Lin, "Vehicle speed detection and identification from a single motion blurred image," in *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, vol. 1, 2005, pp. 461–467.

[4] J. Hu, Y. Li, S. Niu, and X. Meng, "Exposing digital image forgeries by detecting inconsistencies in principal point," in *Computer Science and Service System (CSSS), 2011 International Conference on*, 2011, pp. 404–407.

[5] A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello, "A visual odometry framework robust to motion blur," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009, pp. 2250–2257.

[6] S. Osswald, A. Hornung, and M. Bennewitz, "Learning reliable and efficient navigation with a humanoid," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 2375–2380.

[7] J. M. Hilkert, "Inertially stabilized platform technology concepts and principles," *Control Systems, IEEE*, vol. 28, no. 1, pp. 26–46, 2008.

[8] E. Akgul, M. Mutlu, A. Saranli, and Y. Yazicioglu, "A comparative evaluation of adaptive and non-adaptive sliding mode, lqr amp; pid control for platform stabilization," in *Control Applications (CCA), 2012 IEEE International Conference on*, 2012, pp. 1547–1552.

[9] M. Smith, A. Boxerbaum, G. Peterson, and R. Quinn, "Electronic image stabilization using optical flow with inertial fusion," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 1146–1153.

[10] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman, "Motion-invariant photography," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 71:1–71:9, Aug. 2008.

[11] M. Poulose, "Literature survey on image deblurring techniques," *International Journal of Computer Applications Technology and Research*, vol. 2, no. 3, pp. 286 – 288, 2013.

[12] S. Schuon and K. Diepold, "Comparison of motion de-blur algorithms and real world deployment," *Acta Astronautica*, vol. 64, no. 1112, pp. 1050 – 1065, 2009.

[13] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007.

[14] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 30:1–30:9, July 2010.

[15] S. Nayar and M. Ben-Ezra, "Motion-based motion deblurring," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 689–698, 2004.

[16] A. Agrawal, M. Gupta, A. Veeraraghavan, and S. Narasimhan, "Optimal coded sampling for temporal super-resolution," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 599–606.

[17] M. Tico, N. Gelfand, and K. Pulli, "Motion-blur-free exposure fusion," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 3321–3324.

[18] P. Favaro and S. Soatto, *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[19] A. Basu and K. Ravi, "Active camera calibration using pan, tilt and roll," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 27, no. 3, pp. 559–566, 1997.

[20] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 2432–2439.

[21] R. Ferzli and L. Karam, "A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb)," *Image Processing, IEEE Transactions on*, vol. 18, no. 4, pp. 717–728, 2009.