A NOVEL REAL-TIME INERTIAL MOTION BLUR METRIC WITH APPLICATIONS TO MOTION BLUR COMPENSATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET MUTLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2014

Approval of the thesis:

A NOVEL REAL-TIME INERTIAL MOTION BLUR METRIC WITH APPLICATIONS TO MOTION BLUR COMPENSATION

submitted by **MEHMET MUTLU** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Afşar Saranlı
Supervisor, **Electrical and Electronics Engineering Dept., METU** _____

Assoc. Prof. Dr. Uluç Saranlı
Co-supervisor, **Computer Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Abdullah Aydın Alatan
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Dept., METU _____

Assoc. Prof. Dr. Uluç Saranlı
Computer Engineering Dept., METU _____

Assist. Prof. Dr. Yiğit Yazıcıoğlu
Mechanical Engineering Dept., METU _____

Assist. Prof. Dr. Ahmet Buğra Koku
Mechanical Engineering Dept., METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MEHMET MUTLU

Signature :

ABSTRACT

A NOVEL REAL-TIME INERTIAL MOTION BLUR METRIC WITH APPLICATIONS TO MOTION BLUR COMPENSATION

Mutlu, Mehmet
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Afşar Saranlı
Co-Supervisor : Assoc. Prof. Dr. Uluç Saranlı

August 2014, 70 pages

Mobile robots suffer from sensory data corruption due to body oscillations and motion disturbances. In particular, information loss in images captured with on board cameras can be very high, may become irreversible or computationally costly to compensate. In this thesis, a novel method to minimize average motion blur captured by such mobile visual sensors is proposed. To this end, an inertial sensor data based motion blur metric, MMBM, is derived. The metric can be computed in real time. Its accuracy is validated through a comparison with optic-flow based motion-blur measures. The applicability of MMBM is illustrated through a motion blur minimizing system implemented on the experimental SensoRHex hexapod robot platform by externally triggering an on board 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 image based blur metric computed on captured images. MMBM based motion blur minimization system, through an appropriate modulation of the frame acquisition timing, not only reduces average motion blur, but also avoids frames with extreme motion blur, resulting in a promising, real-time motion blur compensation approach.

Keywords: Motion Blur, Inertial Measurement, Image Frame Triggering, Real Time, Dynamical Systems

ÖZ

ÖZGÜN GERÇEK ZAMANLI ATALETSEL HAREKET BULANIKLIĞI ÖLÇEĞİ VE HAREKET BULANIKLIĞI GİDERME ÜZERİNE UYGULAMALARI

Mutlu, Mehmet
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Afşar Saranlı
Ortak Tez Yöneticisi : Doç. Dr. Uluç Saranlı

Ağustos 2014 , 70 sayfa

Mobil robotlar, gövde salınımları ve hareket bozucularından kaynaklanan algılayıcı verilerindeki bozulmadan olumsuz etkilenirler. Özellikle robot üzerinde bulunan kameralar ile kaydedilen resimlerdeki bilgi kaybı çok fazla olabilir ve bu bilgi kaybı geri alınamaz yada resmin bulanıklığını az da olsa giderebilmek işlemsel olarak masraflı olabilir. Bu çalışmada, mobil kameralar ile yakalanan resimlerde oluşan ortalama hareket bulanıklığını azaltacak özgün bir yöntem önerilmiştir. Bu kapsamda, sadece ataletsel veriler kullanılarak gerçek zamanlı hesaplanabilir hareket bulanıklığı ölçeği, HBÖ, türetildi ve optik akış sonuçları ile karşılaştırılarak tutarlılığı onaylandı. HBÖnün kullanışlılığını ifade edebilmek için SensoRHex üzerinde hareket bulanıklığını aza indirgeyecek bir sistem yapıldı. Bu sistemde, düz bir zeminde ileri doğru giden robot üzerindeki bir kamera gerçek zamanlı hesaplanan HBÖnün değerine göre harici olarak tetiklendi. Önerilen sistemle alınan resimlerdeki hareket bulanıklığı sabit aralıklarla alınan resimlerdeki hareket bulanıklığı ile görsel değerlendirerek ve alınan resimler üzerinden hesaplanan bir hareket bulanıklığı ölçeği ile karşılaştırıldı. HBÖ tabanlı ve resim alma anlarının kontrol edilmesine dayalı hareket bulanıklığı azaltan sistemin, ortalama hareket bulanıklığını azaltmakla kalmayıp, aşırı hareket bulanıklığına uğramış resimlerden de kaçınması, ümit vaat eden bir gerçek zamanlı hareket bulanıklığı giderme yöntemi olduğunu ortaya koymuştur.

Anahtar Kelimeler: Hereket Bulanıklığı, Ataletsel Ölçum, Resim Karesi Tetikleme, Gerçek Zamanlı, Dinamik Sistemler

To My Family

ACKNOWLEDGMENTS

First things first, I would like to express my deep gratitude and sincere appreciation to my supervisor Assoc. Prof. Dr. Afşar Saranlı for crafting a competent researcher out of me. I am grateful for his guidance, encouragement and help since I joined his team in RoLab. He has always been a perfect advisor by giving a broad vision, teaching how to tackle with academically valuable problems and considering me as one of his colleagues. I really appreciate his support and the space he has provided me to pursue my own goals. The excellent discussions we had were not only led me on a valuable research track, but also enriched my social life.

I would like to thank Assoc. Prof. Dr. Uluç Saranlı for his extraordinary guidance which extensively improved the quality of my research through out this thesis. I believe ATLAS is one of the rare laboratories where one can get hands on help and guidance from professors. It was an amazing experience to work with Assoc. Prof. Dr. Uluç Saranlı. I would like to thank Asst. Prof. Dr. Buğra Koku, for reminding me about the priorities of life and giving me extremely enjoyable hobbies in addition to the most of my mechatronics knowledge. Prof. Aydın Alatan enlightened me with his fundamental lectures and always gave me a warm welcome in his lab.

I am a lucky person to work with Gökhan Koray Gültekin, Emre Ege, Orkun Öğücü, Babak Rohani, Emre Akgül, Salih Can Çamdere, Çağlar Seylan and Görkem Secer. We finished many collaborative works and they always made the countless hours I spent in the lab enjoyable for me. I am in depth to my home mates Oğuzhan Zilci and Emin Zerman for being excellent buddies.

I am thankful to TUBITAK for partially supporting this research.

Finally, yet the most importantly, I am grateful to my beloved mom, Saffet Mutlu, and dad, Hasan Mutlu, for their limitless support.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

μ	Motion Based Motion Blur Metric (MMBM)
$\mu*$	Aproximated value of MMBM
(x,y,z)	Coordinates of a fixed point on a scene
(X,Y,Z)	Coordinate frame of a camera
ω_x	Rotatinal velocity of a camera on pitch axis
ω_y	Rotatinal velocity of a camera on yaw axis
ω_z	Rotatinal velocity of a camera on roll axis
(u,v)	Coordinates of a pixel on camera sensor
(U,V)	Coordinate frame of a camera sensor

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

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 [31], these body oscillations result from the robot's own locomotory behaviors and are hence unavoidable. The similar oscillatory motions are not limited to robots and can be observed on humans and many other camera carrying platforms.

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 [6], [20], or identifying whether an image is manipulated [11], 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 [28]. In [25], 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 lo-

comotion 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.

Probabilistic estimation of a mobile robot's location in the presence of noisy motion and sensors has been a canonical algorithmic problem for robotics research. The increasing importance of the visual sensor to provide diverse information and features for a variety of tasks, not limited to localization/mapping puts increasing demands on the quality of the image data acquired. Agile legged robot platforms on the other hand pose particular difficulties for the camera sensor since the inherent time-varying motion of the platform results in visual disturbances such as motion blur which can in turn have a devastating effect on automatically computed image features. Popular image features such as edges, Harris corners[9], SIFT[22] and FAST[30] are used as a basis for a large number of estimation and visual perception algorithms. Motion Blur which is a particularly severe motion induced distortion, makes it very difficult and sometimes impossible to reliably compute these image features, resulting in a chain collapse of downstream algorithms. We strongly believe that acknowledging the presence of this distortion and development of algorithmic approaches exploiting the properties of the application to compensate it are an important research direction.

As exemplified by nature, it is our belief that the vision sensor will have a dominant role in the success of future robotic systems. This is also true for legged robots and the importance is amplified particularly because these promise exceptional dexterity and terrain mobility in unstructured environments. We therefore strongly believe in developing theoretical and algorithmic approaches to improve the quality of the visual data collected on a dextrous legged robotic platforms.

1.2 MOTIVATION

Motion blur in moving cameras has always been a problem especially when the movement of camera is very fast and impulsive. While I was doing gait optimization experiments with the six legged robot SensoRHex in RoLab and ATLAS, I tried to estimate

the position of robot online by solving external calibration parameters of an onboard camera. In this approach camera is taking images of a landmarks on the scene whose global world coordinates are known. I was able to calculate the location of robot using the homographic approach in computer vision techniques. However some of the captured images were extremely blurry and information loss was very high that extracting landmarks on the images were impossible. An exaggerated example for the scenario I encountered can be seen in Fig. 1.1(a). There are ways to reduce amount of motion blur. Reducing the exposure time reduces the motion blur since it is directly proportional to the motion blur amount. Longer the sensor accumulates photons falling on a moving camera, higher the motion blur amount. Lower exposure times result in darker images, because, amount of light for one image shot would not be enough. One possible way to have acceptable images with low exposure times is increasing the ambient light. The light conditions may not be controllable for every scenarios. It is still possible to lighten images by multiplying whole pixel readings with a constant gain factor. But, the main problem is the signal to noise ratio. Capturing an image with low exposure rates means noisier images as seen in Fig. 1.1(b). That type noise is usually referred as the salt and pepper noise.

(a) (b)

Figure 1.1: Adjusting camera parameters is not always a solution. (a) Increasing integration time increase SNR but motion blur also increases. (b) Taking images with low exposure time results in low Signal to Noise Ratio (SNR).

The main observation lead to the work done in this thesis is that some of the captured images during gait experiments of SensoRHex were very blurry while some others were quite sharp. Six image samples captured during the straight locomotion

of SensoRHex can be seen in Fig. 1.2. Some of those images are sharp (Fig. 1.2(a), Fig. 1.2(e)) some of them are very blurry (Fig. 1.2(b), Fig. 1.2(d)) and the blur level on the others are moderate. Exploring the reason behind this observation was the starting point of this work. That exploration lead to the derivation of relation between camera motion and the resulting amount of motion blur.

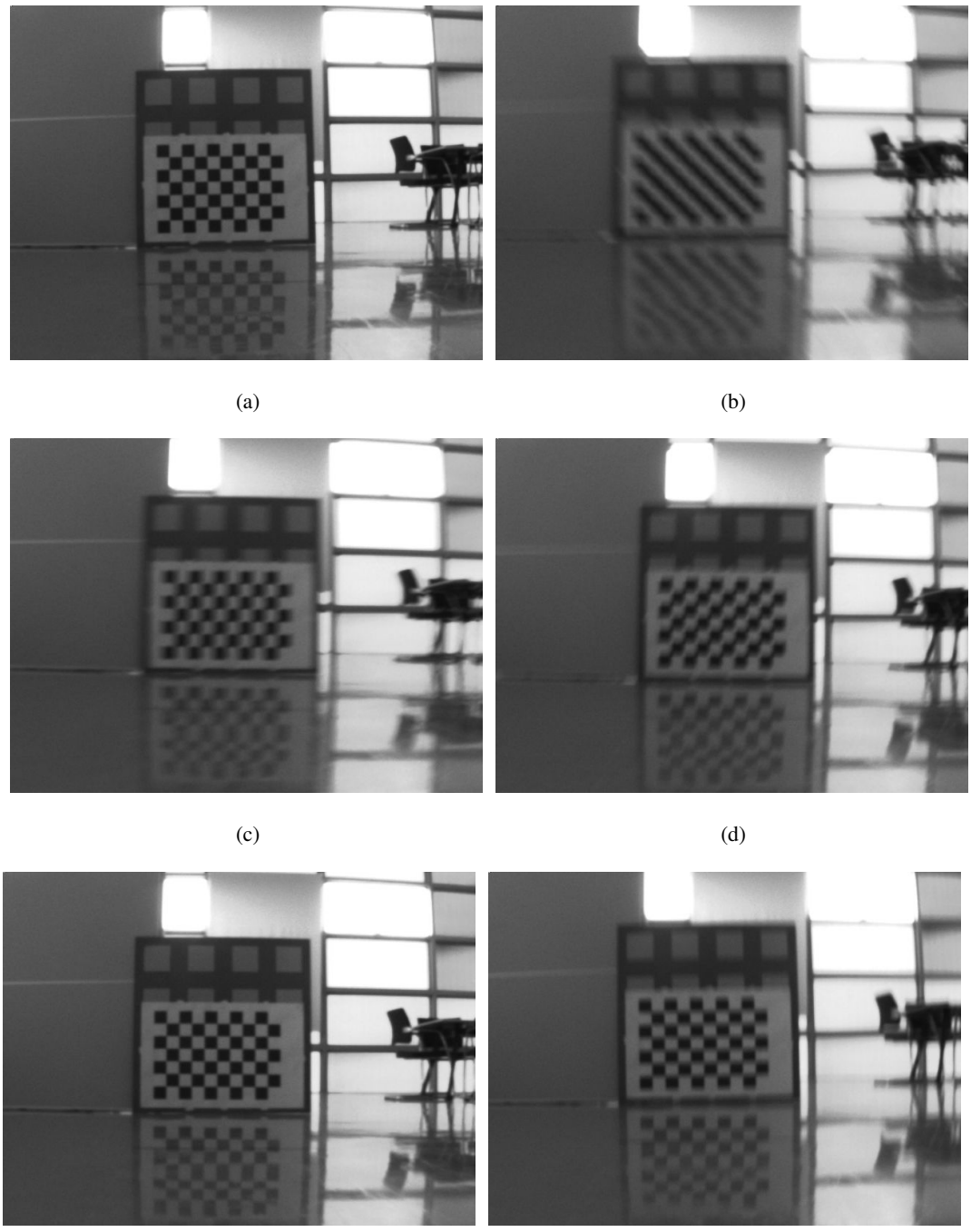
1.3 METHODOLOGY

The main objective of this thesis is to derive and analyse a metric that relates camera motion to the corresponding motion blur in images that will be captured by a moving camera. Although such a metric can be calculated from captured images using image processing techniques, the aim is to have an intuition of how much motion blur there will be on image. Hence, the metric should incorporate sensors other than a camera itself to measure motion of a camera. One of the common ways of tracking motion is using inertial sensors such as accelerometers and gyroscopes. In this study, angular velocity measurements are used to derive the motion blur metric.

An important criteria is that such detection and measurement pre-processing should be computationally inexpensive to be implemented in real-time with minimal delay. Even though, final analytical calculations are not possible, a Riemann Sum approximation is adopted to calculate the value of the metric.

Once such a motion blur metric is obtained, it can be used to track the motion blur of images even before the image is captured. This information can be used to in a variety applications such as detecting an image is extensively blurred or modulating the image acquisition moments by delaying the image capture triggering signal to obtain sharper images. Minimizing corruptive effects of motion blur increases the performance of computer vision algorithms that require feature extraction.

Metric is illustrated in detail by analyses done in MATLAB environment. A hardware setup consisting of a gyroscope, a camera and a microcontroller is created to collect real data on SensoRHex. Further analyses are done using the real data. Moreover, the same hardware setup is used to trigger the camera externally to modulate image acquisition instances depending on the value of real-time calculated metric.



(a) (b)

(c) (d)

(e) (f)

Figure 1.2: Consecutive video frames captured with an onboard camera while SensoRHex is walking on a flat surface. The sequence illustrates interleaved blurred and sharp images.

1.4 CONTRIBUTIONS

The present study focuses on robot platforms with large magnitude, time-varying ego-motion such as the case for dynamically dextrous legged robots. A good example is the popular RHex morphology: a hexapod with exceptional terrain mobility[31]. An example of RHex morphology, SensoRHex, can be seen in Fig. 1.4 which is actively used for experiments in RoLab Other recent examples are dynamically dextrous quadrupeds developed by Boston Dynamics[29]. We consider the most significant motion induced visual degradation, the motion blur, and present an approach to generate a camera data stream with a significantly reduced motion blur content. This is achieved with data from inertial sensors (gyroscopes) and through active control of the camera image acquisition cycle, in particular the frame trigger timing. The primary contributions of the thesis are threefold: Firstly, we propose a novel blur metric based on transformed inertial measurements to track the blur in the image. Secondly, the metric is used to modulate image acquisition instances to avoid extremely blurred images. Finally, we successfully apply this metric in position measurement task where individual frame based measurements are significantly improved, paving the way for improvements in pose estimation problems.

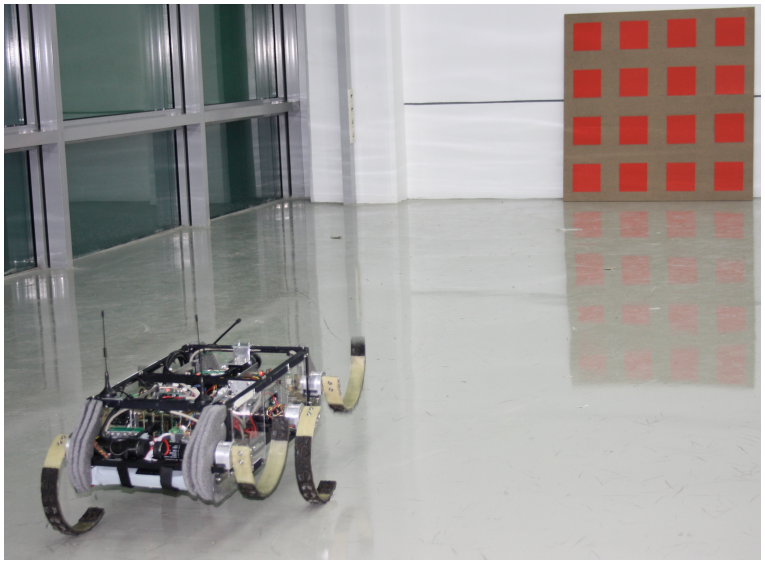


Figure 1.3: SensoRHex is walking on a flat surface.

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 work 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 most fundamental 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.

1.5 OUTLINE OF THE THESIS

The thesis is organized as follows: Chapter 2 begins by presenting relevant literature for our study, followed by Section 3 where our new motion blur metric based on angular velocity measurements and all camera motion measurements are presented. Chapter 4 provides further analysis on sensitivity of metric for all parameters used in the metric derivation and how to calculate the metric in real time. Chapter 5 gives discussions on different possible ways on how to use the proposed metric and details of the usage of metric for image acquisition. Experimental results for obtaining sharper images and the implication of those images are presented in chapter 6. Finally, chapter 7 gives the concluding remarks.

CHAPTER 2

LITERATURE REVIEW

Motion blur is a very common problem that many researchers attacked to the problem from many different perspectives. The proposed solutions in the literature can be divided into two main groups in terms of their approach to deal with motion blur. Software based approaches focus on removing the motion blur from images after it is captured. Some other approaches are also considered in software methods, even though, they may make use of complementary sensors and hardware and their techniques to remove the motion blur may significantly differ from each other. On the other hand, hardware approaches mainly try to stabilize the camera motion when images are desired to be captured with a mobile camera.

2.1 MOTION BLUR REMOVAL; SOFTWARE BASED POST PROCESSING

Software based motion blur removal techniques primarily focus on individual frames only after an image is captured with motion blur [27]. Many of them use only single image to resharpen it. The downside of single frame software methods is that the deconvolution operation is ill-defined since some of the information is permanently lost due to the nature of motion blur. Although, fast deblurring algorithms that can be executed online [3], deconvolution techniques are usually computationally costly as well and may be difficult to implement in real-time applications [32].

Inertial and visual sensors can act as complementary pairs [5], with inertial sensors used to obtain extra information on motion. Using inertial sensors the Point Spread Function (PSF) can be estimated. PSF is a kernel which describes the motion blur and

gives the blurred image when convolved with a sharp one. PSF knowledge is essential for deblurring. It can be estimated with inertial sensors [13] or with a complementary camera [24]. Hence, "non-blind" deconvolution can be applied for deblurring as a better alternative to blind deconvolution. Deconvolution on image domain is not well defined like the convolution operation. Common implementation of deconvolution involves search of sharp image in pixel domain. In yet another application, a camera is precisely moved in a carefully designed way to modify the PSF to increase performance of motion deblurring with deconvolution [19]. The downside of this approach is that first it needs to blur whole image even if the camera is not mounted on a mobil platform and only motion blur caused by a moving object is desired to be removed. Although, it can successfully remove most of the motion blur, there remains some artifacts on the background scene. Also, estimating motion from inertial sensors makes a good initial guess for parameter estimation for deblurring or feature extraction algorithms [14].

Camera shutter hardware control, is also used in certain applications to compensate for motion blur. The simplest solution is limiting exposure time, but the image SNR decreases due to the reduced amount of light integration. Special lighting is usually required for this approach to be successful. Light integration pattern can be manipulated to minimize deconvolution noise [1] by using coded exposure technique. Fusion of frames which are exposed at different durations on the same scene can also be used to reduce motion blur [39]. However, the computational complexity is a considerable burden of these approaches especially when they are used on low power autonomous mobile robots.

2.2 MOTION BLUR AVOIDANCE; HARDWARE SOLUTIONS

Hardware methods are widely used by camera manufacturers. High end commercial cameras use lens or sensor motion techniques for image stabilization. The theory behind lens stabilization technique is moving lens [35] in a plane to ensure photons emitted or reflected from fixed objects fall upon the same region of camera sensor. The other method also uses the same idea, but, lens remains stationary and sensor is moved to compensate the motion of camera [17].

Another commonly used hardware solution is mounting the camera on top of a stabilization platforms. Gimballed platforms, have rotary structures that are connected to each other with actuators and they can cancel the rotations exhibited on main body such that camera would not rotate [18]. Stewart platforms and their variants are commonly used for line-of-sight stabilization [10], [2]. Stewart platform is a well analyzed and commonly used mechanism to move a platform in both 3D rotation and 3D translation. Thus, it can move to all positions and orientations in its obstacle-free workspace. Such platforms are usually mechanically complex, costly and often require advanced control algorithms. Moreover, robust pose estimation is required for stabilization platforms, but, a challenging problem for highly dynamic robots due to inertial measurement drift [33]. It may be difficult to use Stabilization platforms on small scale robotic platforms due to their size, weight and effects on robot dynamics. However, size and weight issues can be solved by designing a custom stabilization platforms. For example a stabilization platform that is designed to be used on mobile robot applications can be seen in Fig. 2.2. As an alternative to mechanical stabilization platforms optic flow based stabilization can be used [12].

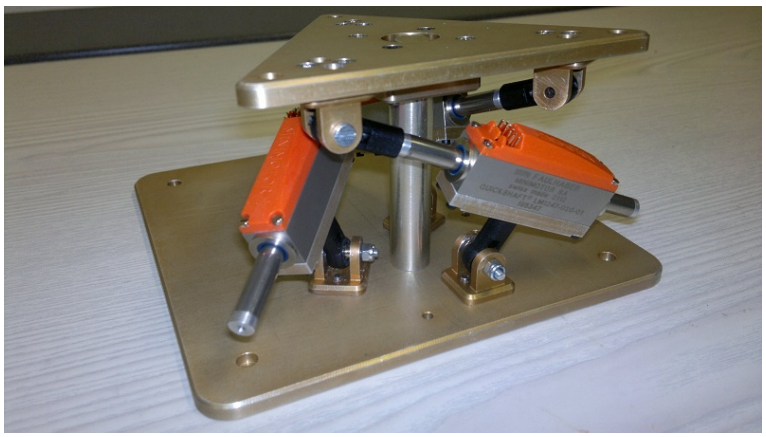


Figure 2.1: 3D rotary motion stabilizing platform designed in RoLab.

2.3 OUR METHOD

Our approach to tackle motion blur involves 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 setup to reduce average motion blur while capturing a video sequence.

Although there were attempts to map motion of camera to the motion blur formation [26], there is a lack of motion blur metric in the literature. One of the techniques seen the literature uses the magnitude of an acceleration vector obtained from a MEMS bases inertial measurement unit to detect whether a barcode reader is moving or not [21]. Even inclenometers are used to detect whether camare moves or not [40]. The approaches only tries to detect whether motion exists. Actual motion blur requires the measurement of actual motion.

In order to minimize average motion blur, our proposed method is triggering the camera at suitable time instances. Some conceptual ideas about having a motion blur metric are discussed in [36]. One of the implementations give a fixed delay if motion is detected [16]. A more advanced one also limits maximum exposure time if motion blur is detected [34]. The possibility of terminating image capture is claimed in [37]. One of the promising applications that exploit oscillatory behaviour of camera is presented in [41]. Their approach is estimating the position of camera and capturing images while camera is passing thorough the same locations such that jitter caused by the oscillatory motion of camera can be avoided. Our study has many common discussions with the commercially applied techniques, yet, we address unanswered parts in those works.

CHAPTER 3

INERTIAL MEASUREMENT BASED MOTION BLUR METRIC DERIVATION

Exact motion blur that occurs due to robot ego motion can only be extracted by measuring translation and rotation of camera throughout integration time, namely the duration for which camera integrates light [7], [4], [24]. The focus of this thesis is on estimating the average magnitude of motion blur on the image plane at any given time such that sharper images can be obtained by triggering the camera at favorable time periods. The primary observation that enables such an opportunity is that a legged robot has a considerable inertia which will force the robot to have attitude oscillations with a predictable low-pass character. Hence, there will be repeating time instances where the body velocity will be small and these times can be predicted short-time in advance.

The major causes of motion blur are angular and translational velocities of a camera around three axis. Motion blur can also be caused by moving objects in a stationary scene, an unstationary scene or the zoom event of a camera. In order to be able to analyze the complete motion blur all of the aforementioned causes should be known or they should be extractable from an image or sequence on images. It was already explained that whole motion blur can mostly be extracted from captured images using image processing techniques. Even though, knowing the motion blur after an image is already blurred can be useful to resharpen it, the aim in this work is to predict the motion blur before capturing an image. In this context, scene, objects in the scene and zoom factor of the camera are assumed to be fixed and only the motion blur caused by the egomotion of a camera is aimed to be avoided by exploiting quasi-periodic

behavior of walking.

Derivation of a metric to estimate the motion blur amount in an image that will be captured requires the instantaneous rotational and translational velocity measurements of a camera. Projected locations of fixed world points on the image sensor depends on only the relative angle of those points with respect to the pinhole when camera undergoes only the rotation motion. Rotational velocity can be directly measured with a gyroscope. In the translation case, finding projected velocities of fixed world points requires the depth of scene information in addition to the pure linear velocity of the camera. Measuring pure translational velocity is not as straight forward as obtaining the rotational velocities. There are two common methods. The first one involves measuring the linear acceleration of the camera with an accelerometer and taking its integral. However, commercially available accelerometers have a certain error characteristics and taking the integral of those errors is likely to drift the integrated velocity measurements. The second method is tracking the location of the camera with a camera based external tracking system and extracting the velocity of the robot by taking the derivative of the position. Although, the latter method does not accumulate errors, the derivative operation is more susceptible to measurement variations caused by noise. Furthermore, including the translational motions to the metric requires the depth of scene information which has to be measured with a depth sensor such as kinect, laser scanner or an external tracking system. Another observation is that, when camera is used in a large workspace, the depth of scene is usually quite large. Translational variations of a camera is less dominant when the scene is far away form the camera. Due to the dominance of rotational motions on the motion blur formation and ease of real implementation, first the motion based motion blur metric will be derived by considering only the rotational motion of the camera and the effects of translational motion will be analyzed later on.

3.1 ROTATION MOTION BASED MOTION BLUR METRIC (MMBM)

MMBM derivation for pure rotational motion consists of modeling the motion blur formation process on the image plane and computing a scalar metric out of it. In the present case, MMBM only considers instantaneous angular velocities, transforming

them through the projective transformation onto the image plane rather than considering the entire integration time [23]. If the rotational velocities remain constant during exposure time, MMBM becomes a better approximation to the actual motion blur.

3.1.1 Notation

In the pure rotational scenario, there is a camera pointing through a stationary scene and the camera rotates in all there axes. The variables used on MMBM derivation are illustrated in Fig. 3.1. World and image coordinate frames, a fixed point on the real world, its projection on the image plane, rotational velocities and focal length of the camera are denoted by (X,Y,Z), (U,V), (x,y,z), (u,v), (w_x, w_y, w_z) and f respectively.

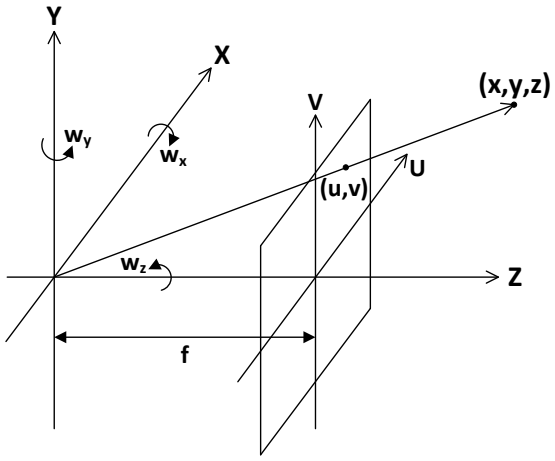


Figure 3.1: Illustration of frames and definitions used in the derivation of the MMBM.

3.1.2 Average Flow Derivation

The main principle behind deriving a motion blur metric is deriving the velocity transformation from camera motion to the projected points motion on image sensor plane. Once the relations are obtained the desired metric will be the average of instantaneous variations on image sensor.

3.1.2.1 Camera Model

A basic pinhole camera model is used for modeling the camera projection.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f\dfrac{x}{z} \\ f\dfrac{y}{z} \end{bmatrix} \tag{3.1}$$

Respectively, u and v represents x and y coordinates of image plane. In homogenous coordinates, basic pinhole camera matrix, P, can be written as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f\dfrac{x}{z} \\ f\dfrac{y}{z} \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.2}$$

Now, the relation between world coordinates and image plane coordinates is known. Inverse camera model assumes Z is known and it turns out to be as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} z\dfrac{u}{f} \\ z\dfrac{v}{f} \\ z \end{bmatrix}. \tag{3.3}$$

3.1.2.2 Intrinsic and Extrinsic Camera Calibration Matrices

In the metric derivation, camera is assumed to be pre-calibrated, so, intrinsic camera parameters such as distortion are not included in the final form of the metric since the aim was to keep the metric as simple as possible. But, the camera model can be improved further by considering finite projective camera model and camera calibration matrices.

Intrinsic calibration matrix, K, can easily be used instead of the basic pinhole camera matrix P.

$$K = \begin{bmatrix} a_x & s & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

In the K matrix, $a_x = f * m_x$; $a_y = f * m_y$; $x_0 = m_x * p_x$ and $y_0 = m_y * p_y$. m_x and m_y represents number of pixels per unit distance in image coordinates. p_x and p_y shows principal point offset of a camera. Finally, s is the skew parameter. Exact motion blur on images are actually a parameter of all distortions. However, tracking only the main trends will be sufficient for the derivation of a general purpose metric. Moreover, images can be calibrated independently and the metric derived without considering the intrinsic calibration would work.

External calibration gives camera frame location and orientation with respect to a known world frame as shown in Fig. 3.2. The extrinsic calibration matrix is calculated as,

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \\ 1 \end{bmatrix} = \begin{bmatrix} R & -R\widetilde{C} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{3.5}$$

In this study, extrinsic calibration is also not needed. Camera can be simply assumed to be at origin of the world coordinate frame.

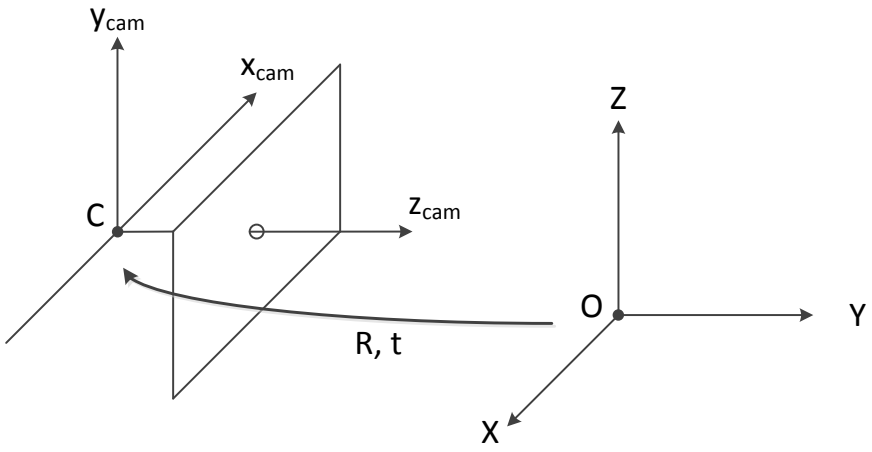


Figure 3.2: Illustration of extrinsic camera calibration.

3.1.2.3 Velocity Relation

The metric that will be derived aims to predict the motion blur. The exact camera motion will not be available prior to the image capture, but, the rotational velocity of

the camera can be measured and the projected velocity of fixed world points on the image sensor can be extracted. Time integration of projected points velocity on image sensor, would give the exact motion blur. Under the assumption that camera velocity will remain constant during the exposure period, the velocity of the projected points will give a relatively accurate representation of the motion blur. Time derivative of the projected points on the image coordinates can be found by taking the derivative of the pinhole camera model:

$$\frac{d}{dt}\begin{bmatrix} u \\ v \end{bmatrix} = f\begin{bmatrix} \frac{\dot{x}z - x\dot{z}}{z^2} \\ \frac{\dot{y}z - y\dot{z}}{z^2} \end{bmatrix} = \begin{bmatrix} \frac{f}{z} & 0 & -\frac{fx}{z^2} \\ 0 & \frac{f}{z} & -\frac{fy}{z^2} \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \tag{3.6}$$

For the name convention following matrix will be called as L.

$$\begin{bmatrix} \frac{f}{z} & 0 & -\frac{fx}{z^2} \\ 0 & \frac{f}{z} & -\frac{fy}{z^2} \end{bmatrix} = L \tag{3.7}$$

Rotation of the camera with respect to its focal point on a stationary environment is analogous to rotating whole world around a fixed camera in terms of mathematical derivations. The latter approach can be easier to visualize. The only difference is the sign of the rotation matrix w. The velocity of a point in the scene when the scene is rotating around the fixed camera with respect to an arbitrary vector, w, that passes through the focal point of the camera is defined as

$$\dot{P} = W \times P, \tag{3.8}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{3.9}$$

3.1.2.4 Definition of Rotation Motion Based Motion Blur Metric (MMBM)

Obtaining the averaged optic flow that is caused by the rotary egomotion of a camera requires the integration of instantaneous image plane optic flow vector magnitudes

caused by camera rotation at inertial measurement time instance. This leads to the definition of MMBM in vector format as

$$\mu := \frac{1}{\Delta u \Delta v} \iint_{u_{min},v_{min}}^{u_{max},v_{max}} \left\| \begin{matrix} \dot{u} \\ \dot{v} \end{matrix} \right\| \mathrm{d}v\mathrm{d}u. \tag{3.10}$$

(3.11) can be written in open form

$$\mu := \frac{1}{\Delta u \Delta v} \iint_{u_{min},v_{min}}^{u_{max},v_{max}} \sqrt{\dot{u}^2 + \dot{v}^2}\, \mathrm{d}v\mathrm{d}u, \tag{3.11}$$

where Δu and Δv are defined as $(u_{max} - u_{min})$ and $(v_{max} - v_{min})$ respectively. μ can be used to track the average motion blur and from now on it will be the rotary motion based motion blur metric (MMBM). As seen in (3.10) MMBM does not involve any time integration. It calculates instantaneous average magnitude of the projected points' velocity vectors.

Inserting (3.6), (3.9) and (3.3) respectively into instantaneous optic flow vector in (3.10) to be able to explicitly evaluate the integral, the following relation is obtained,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{f}{z} & 0 & -\frac{u}{z} \\ 0 & \frac{f}{z} & -\frac{v}{z} \end{bmatrix} \begin{bmatrix} 0 & w_z & -w_y \\ -w_z & 0 & w_x \\ w_y & -w_x & 0 \end{bmatrix} \begin{bmatrix} z\frac{u}{f} \\ z\frac{v}{f} \\ z \end{bmatrix}. \tag{3.12}$$

Negative rotation matrix is used in (3.12) since the world point is stationary and camera is rotating in the problem. The relation given in (3.12) can be evaluated to obtain $\dot{u}$ and $\dot{v}$,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{f}{z} & 0 & -\frac{u}{z} \\ 0 & \frac{f}{z} & -\frac{v}{z} \end{bmatrix} \begin{bmatrix} \frac{zw_z}{f}v - w_y z \\ -\frac{zw_z}{f}u + w_x z \\ \frac{zw_y}{f}u + \frac{zw_x}{f}v \end{bmatrix} = \begin{bmatrix} +w_z v - w_y f - \frac{w_y}{f}u^2 + \frac{w_x}{f}uv \\ -w_z u + w_x f - \frac{w_y}{f}uv + \frac{w_x}{f}v^2 \end{bmatrix}. \tag{3.13}$$

After obtaining $\dot{u}$ and $\dot{v}$ separately, euclidian norm of the projected points velocity vector can easily be found;

$$\left\| \begin{matrix} \dot{u} \\ \dot{v} \end{matrix} \right\| = \sqrt{\dot{u}^2 + \dot{v}^2}. \tag{3.14}$$

Individual components of (3.14), $\dot{u}$ and $\dot{v}$, are evaluated as

$$\begin{aligned}
\dot{u}^2 =&(+w_z v - w_y f - \frac{w_y}{f}u^2 + \frac{w_x}{f}uv) * (+w_z v - w_y f - \frac{w_y}{f}u^2 + \frac{w_x}{f}uv)\\
=&w_z^2v^2 - w_z w_y f v - \frac{w_z w_y}{f}u^2 v + \frac{w_z w_x}{f}uv^2 - w_y w_z f v + w_y^2 f^2 + w_y^2 u^2\\
&- w_y w_x uv - \frac{w_y w_z}{f}u^2 v + w_y^2 u^2 + \frac{w_y^2}{f^2}u^4 - \frac{w_y w_x}{f^2}u^3 v + \frac{w_x w_z}{f}uv^2\\
&- w_x w_y uv - \frac{w_x w_y}{f^2}u^3 v + \frac{w_x^2}{f^2}u^2 v^2\\
=&\frac{w_y^2}{f^2}u^4 - 2\frac{w_x w_y}{f^2}u^3 v + \frac{w_x^2}{f^2}u^2 v^2 - 2\frac{w_y w_z}{f}u^2 v + 2w_y^2 u^2 + 2\frac{w_x w_z}{f}uv^2\\
&- 2w_x w_y uv + w_z^2 v^2 - 2w_y w_z f v + w_y^2 f^2,
\end{aligned} \tag{3.15}$$

and

$$\begin{aligned}
\dot{v}^2 =&(-w_z u + w_x f - \frac{w_y}{f}uv + \frac{w_x}{f}v^2) * (-w_z u + w_x f - \frac{w_y}{f}uv + \frac{w_x}{f}v^2)\\
=&w_z^2u^2 - w_z w_x f u + \frac{w_z w_y}{f}u^2 v - \frac{w_z w_x}{f}uv^2 - w_x w_z f u + w_x^2 f^2 - w_x w_y uv\\
&+ w_x^2 v^2 + \frac{w_y w_z}{f}u^2 v - w_y w_x uv + \frac{w_y^2}{f^2}u^2 v^2 - \frac{w_y w_x}{f^2}uv^3 - \frac{w_x w_z}{f}uv^2\\
&+ w_x^2 v^2 - \frac{w_x w_y}{f^2}uv^3 + \frac{w_x^2}{f^2}v^4\\
=&\frac{w_x^2}{f^2}v^4 - 2\frac{w_x w_y}{f^2}uv^3 + \frac{w_y^2}{f^2}u^2 v^2 - 2\frac{w_x w_z}{f}uv^2 + 2w_x^2 v^2 + 2\frac{w_y w_z}{f}u^2 v\\
&- 2w_x w_y uv + w_z^2 u^2 - 2w_x w_z f u + w_x^2 f^2.
\end{aligned} \tag{3.16}$$

Then the open form of $\dot{u}^2 + \dot{v}^2$ turns out to be

$$\begin{aligned}
\dot{u}^2 + \dot{v}^2 =&\frac{w_y^2}{f^2}\boldsymbol{u^4} + \frac{w_x^2}{f^2}\boldsymbol{v^4} - 2\frac{w_x w_y}{f^2}\boldsymbol{u^3 v} - 2\frac{w_x w_y}{f^2}\boldsymbol{u v^3} + \frac{w_x^2 + w_y^2}{f^2}\boldsymbol{u^2 v^2}\\
&+ (2w_y^2 + w_z^2)\boldsymbol{u^2} + (2w_x^2 + w_z^2)\boldsymbol{v^2} - 4w_x w_y \boldsymbol{uv} - 2w_x w_z f \boldsymbol{u}\\
&- 2w_y w_z f \boldsymbol{v} + (w_x^2 + w_y^2)f^2.
\end{aligned} \tag{3.17}$$

Although, (3.17) has a neat structure, (3.11) cannot be analytically evaluated due to the square root that is inside of the double integral. A common approach that is used in optimization techniques is taking the square of the euclidian norm in (3.11) thus integral can be analytically solvable. However the aim of MMBM is to model the motion blur from the rotational velocities of camera as accurately as possible. Therefore, (3.11) will be evaluated with numerical methods such as ode45.

3.1.3 Visualization of MMBM

Fixed points in the scene are projected onto the image plane through the camera model. Projected points trace a certain trajectory on the image plane when camera undergoes a rotation around its optical center. Our proposed metric takes into account only instantaneous velocities of projected points. MMBM is actually a continuous integral throughout image plage as stated in (3.11). But, drawing down sampled image plane velocity vectors helps to understand the mechanics of MMBM. Fig. 3.3 illustrates projected velocity vectors for some uniformly sampled function evaluation points throughout the image plane when camera rotates. 640x480 pixel sensor plane having unit pixel size is 4.65μm is represented in metric units.

In Fig. 3.3(a) camera is subjected to 1.0rad/sec rotation in only w_x. In order to give an intuition about how much blur that rotation would cause on an image that is captured in 50ms, the resulting vector sizes are multiplied with to 1000/50 before plotting them on Fig. 3.3. Projections of fixed world points are moving up on the image plane. Magnitudes of those vectors get slightly larger around the upper and lower edges of the sensor plane. Also the direction of movements slightly change around right and left edges since the sensor is planar. Although, all commercial image sensors are currently planar, hemispherical ones that mimics human eye [15] are also likely to appear in the coming years. If the image sensor was hemispherical, projected velocity vectors caused by camera rotation in w_x axis could have been perfectly linear. Fig. 3.3(b) shows the velocity vectors of projected points when camera undergoes 2.0 rad/sec roll motion which is rotation in w_z direction. Compared to rotation in yaw and pitch axes, rolling motion has completely different effect on motion blur. In the pure rolling case, center of the image would never have any motion blur. Even the fastest points, which are around the edges of sensor plane, are a number of times slower than average magnitude of velocity vectors observed in the same yaw and pitch rotation speeds. Please note that w_z in Fig. 3.3(b) is 2 times faster compared to w_x in Fig. 3.3(a). Hence, roll has less significant effect on motion blur compared to pitch and yaw. Roll, yaw and pitch components can be independently considered and added on top of each other to get cumulative result. Fig. 3.3(c) shows the resulting velocity vectors when camera is rotated in yaw and roll axes and Fig. 3.3(d) illustrates the

resulting vectors when camera rotates in all axes at the same time. MMBM calculated from a single gyro reading is magnitude average of vectors as shown in Fig. 3.3 for four different rotation case.

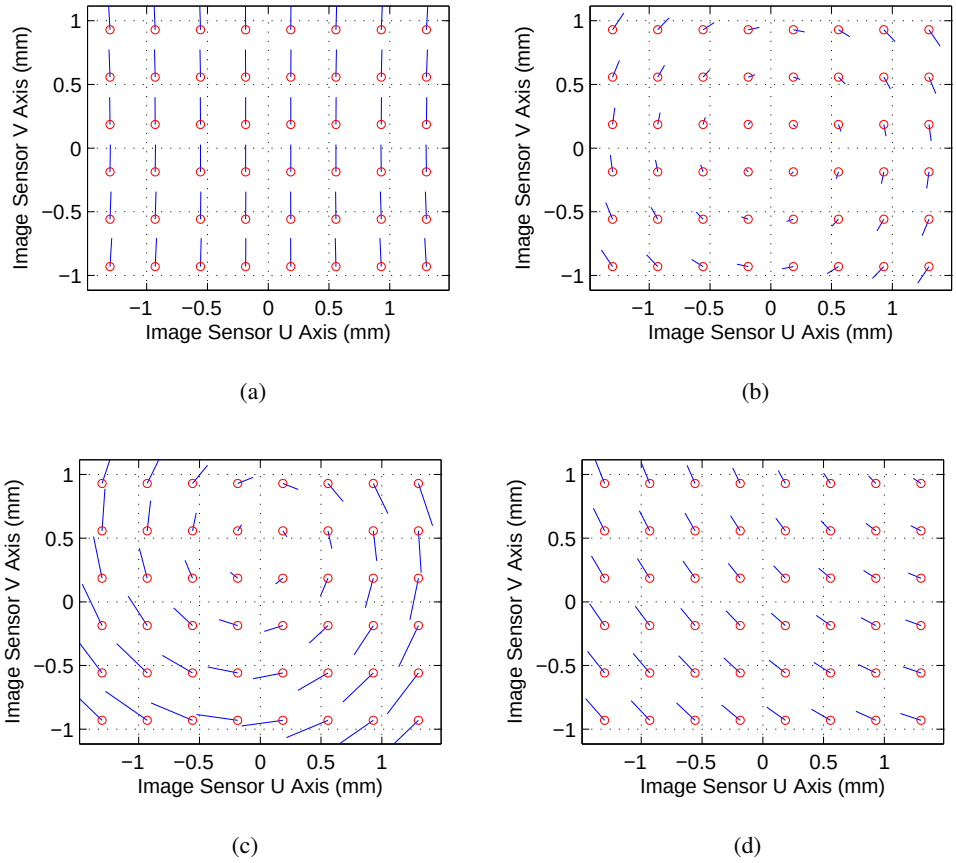


Figure 3.3: Velocity vectors of projected points on image sensor subjected to different camera rotational velocities. Rotational velocity vectors of the camera, $[w_x\ w_y\ w_z]$ in rad/sec, (a) [1.0 0 0], (b) [0 0 2.0], (c) [0 0.5 5.0], (d) [0.5 0.5 1.0].

Gyro data can be collected at high frequencies while robot is moving and MMBM can be calculated for each gyro measurement. Fig. 3.4 illustrates the calculated value of MMBM for each 3D gyro measurement for a slightly longer duration than 2 seconds. It can be seen that rotation in z axis has lesser effect on motion blur formation when compared to rotations in x and y axes.

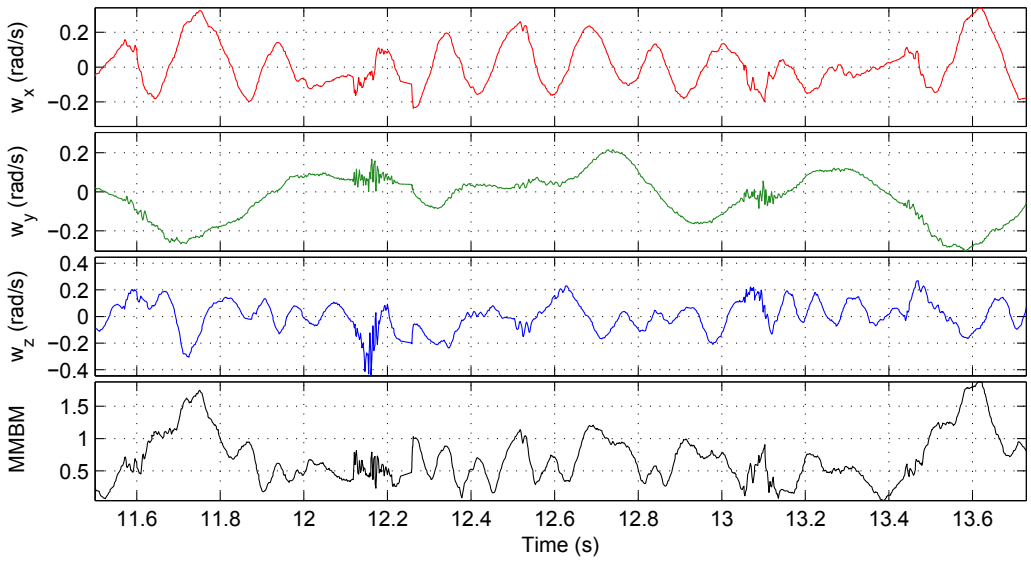


Figure 3.4: Variation of MMBM compared with individual rotational velocities while SensoRHex is walking on a flat surface.

3.2 CAMERA ROTATION AND TRANSLATION MOTION BASED MOTION BLUR METRIC (tMMBM)

In the previous section, a metric to track motion blur amount caused by the rotation motion of the camera was derived and explained. The assumption was that translation would have very little affect on motion blur formation when the robot is working in a large and free space. In other words, objects that the camera is observing will be at least a few meters away from it. Furthermore, the difficulty of measuring translational velocities of a mobile robot was much more challenging than measuring the rotational velocities. In this section, the translation motion of the camera will be added to the derivations done in the previous section. Then, the effects of the translation and be discussed and the cases where translation motion should be used will be analyzed.

3.2.1 Notation

In the whole camera motion scenario is an extension to the pure rotational case where camera both rotates and translates in all there axes. The variables used on tMMBM derivation are illustrated in Fig. 3.5. World and image coordinate frames, a fixed point on the real world, its projection on the image plane, focal length, rotational and

translational velocities of the camera are denoted by (X,Y,Z), (U,V), (x,y,z), (u,v), f, (w_x, w_y, w_z) and (v_x, v_y, v_z) respectively.

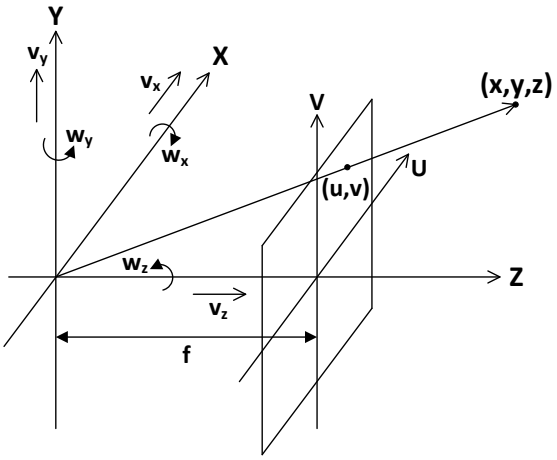


Figure 3.5: Illustration of frames and definitions used in the derivation of the tMMBM.

3.2.2 Motion Based Motion Blur Metric Derivation

Approach in the derivation of tMMBM is pretty much similar to the derivation of MMBM. The same assumptions, e.g. not including intrinsic and extrinsic calibration matrices etc., also hold for the tMMBM. The definition of the tMMBM is still the same as the definition of MMBM, which was shown in (3.11). The only difference is that camera can also freely translate in the current derivation. Hence, the derivation of tMMBM will be the same until (3.12). Only the velocity vector of camera changes. As explained in Section 3.1.2.3 motion of camera is analogous to the motion of scene. In other words, in a setup where camera is moving in a stationary environment, camera can be considered to be stationary and the whole scene, or simply the point of interest on the scene, can be considered to be moving with the same speed which camera should move, only through the reverse direction of the camera. The velocity vector of the scene while camera is moving turns out to be

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{3.18}$$

where v_x, v_y and v_z are the translational velocities of camera in free space.

Inserting the translational motion of the camera to the (3.13) yields the following projected velocities of the scene:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{f}{z} & 0 & -\frac{u}{z} \\ 0 & \frac{f}{z} & -\frac{v}{z} \end{bmatrix} \begin{bmatrix} 0 & w_z & -w_y \\ -w_z & 0 & w_x \\ w_y & -w_x & 0 \end{bmatrix} \begin{bmatrix} z\frac{u}{f} \\ z\frac{v}{f} \\ z \end{bmatrix} + \begin{bmatrix} \frac{f}{z} & 0 & -\frac{u}{z} \\ 0 & \frac{f}{z} & -\frac{v}{z} \end{bmatrix} \begin{bmatrix} -v_x \\ -v_y \\ -v_z \end{bmatrix}. \tag{3.19}$$

When (3.19) is further evaluated the following open form is obtained,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} +w_z v - w_y f - \frac{w_y}{f}u^2 + \frac{w_x}{f}uv \\ -w_z u + w_x f - \frac{w_y}{f}uv + \frac{w_x}{f}v^2 \end{bmatrix} + \begin{bmatrix} -\frac{f}{z}v_x + \frac{u}{z}v_z \\ -\frac{f}{z}v_y + \frac{v}{z}v_z \end{bmatrix}. \tag{3.20}$$

The most striking changes between MMBM and tMMBM are the dependency of scene depth information and requirement to measure translational velocities of a camera as shown in the last matrix of (3.20). Both of them can be measured with sensors, however, they increase the complexity of the motion blur detection system.

After obtaining $\dot{u}$ and $\dot{v}$ of tMMBM separately, euclidian norm of the projected points velocity vector was given in (3.14) and individual components of (3.14), $\dot{u}$ and $\dot{v}$, can be evaluated and the resulting forms will involve a two variable integral that cannot be solved analytically, but still can be evaluated with numerical methods. Instead of explicitly calculating individual components of the euclidian norm that is given in the tMMBM definition, first , calculating numerical values of $\dot{u}$ and $\dot{v}$ and, then, calculating the euclidian norm is less computationally expensive.

3.2.3 Visualization of tMMBM

Any motion of camera causes displacement of projected stationary world points on camera sensor. It is valid for both rotational and translational motion of the camera. Although the rotational motion is scene depth independent, the motion blur induced by the translational motion highly depends on the distance between the camera and the scene.

Individual optical flow vector samples taken at predetermined sensor pixels for a translating camera are illustrated in Fig. 3.6. In this figure, camera is assumed to be pointing towards a flat surface that is perpendicular to the roll axis of a camera and the distance between camera and the surface in the scene is assumed to be 0.5m. The camera properties are the same as the camera model used to illustrate MMBM. Camera resolution is set to 640x480 pixels and each pixel has square shape with 4.65μm edge size.

In Fig. 3.6(a) camera is subjected to 0.5m/sec translation in only w_x. In order to give an idea about an image captured with 50ms exposure time the resulting vectors are scaled with 50/1000. Since camera is moving linearly on x axis only, the resulting vectors have the same magnitude and direction throughout the camera sensor plane. The similar behaviour is also observed in the y axis as seen in Fig. 3.6(a). The only difference is the direction of the optical flow vectors. The magnitude of optical flow vectors are relatively comparable to the ones examined in the MMBM case. However, the main reason is the distance of the scene. All rotational and translational velocities that are used to visualize optical flow vectors are chosen slightly larger than maximum values observed on SensoRHex while it was walking in the fastest mode on flat concrete surface. But the scene depth is usually much deeper than 50cm that is used in Fig. 3.6. Deeper the the scene goes, smaller the optical flow vectors, that are caused by the translation motion, become. Hence, the motion blur caused by the translation motion remains bounded at negligibly small values. But still, having comparable sized vectors is better in terms of illustration purposes.

The affect of approaching or getting away from the surface that the camera is pointing through has a completely different characteristic. The type of motion blur that it creates resembles the zooming related motion blur. Fig. 3.6(c) shows the optical flow vectors when camera is 50 cm away from the surface that it is pointing and camera is moving away from the scene. More specifically, camera is moving at -2m/s on z axis. Similar to the roll motion of the camera, the pixels in the middle of the image suffer less motion blur whereas the ones near the edge of the camera sensor exhibit more variation, i.e. larger optical flow vectors. It is noteworthy to notice that the effect of motion along the z axis has recessive effect on motion blur. Even though the camera in Fig. 3.6(c) moves four times faster than the camera in Fig. 3.6(a) and Fig. 3.6(a).

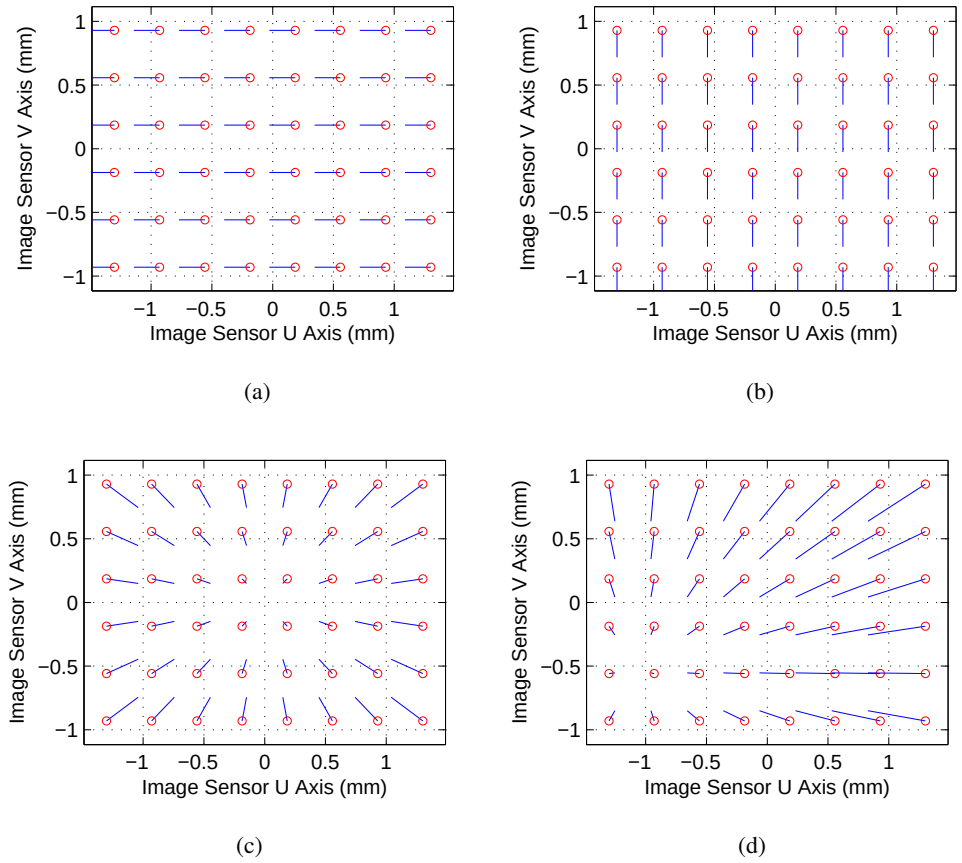
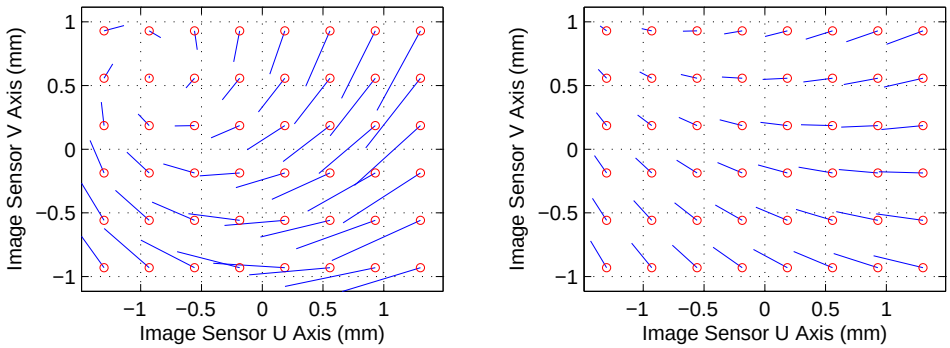


Figure 3.6: Velocity vectors of projected points on image sensor subjected to different camera translational velocities when camera is 0.5m away from the scene. Translational velocity vectors of the camera, $[v_x\ v_y\ v_z]$ in m/sec, (a) [0.5 0 0], (b) [0 0.5 0], (c) [0 0 -2], (d) [0.5 0.25 -2].

The combined motion blur can be decomposed into individual components in terms of motion causing the blur along each axis. All six degrees of motion can be calculated separately and the combined motion blur can be calculated by taking the sum of all optical flow vectors. Fig. 3.6(d) shows a sample of optical flow vector set that is caused by pure translational motion in all three axes.

The cumulative motion blur for both translation and rotation can be very complex. For example, Fig. 3.7(a) shows the corresponding optical flow vectors for the combined motion shown in Fig. 3.3(d) and Fig. 3.6(d). Scene depth is kept 50 cm. Similarly, the Fig. 3.7(b) is the result of combined motion from rotational velocity of Fig. 3.3(d) and half of the translational velocity in Fig. 3.6(d). Although, motion in one of the axes will dominate the others in most of the cases, motion blur such as Fig. 3.7(a) and Fig. 3.7(b) may occur.



(a) (b)

Figure 3.7: Velocity vectors of projected points on image sensor subjected to all type of camera motion. Velocity vectors of the camera, $[\omega_x\ \omega_y\ \omega_z\ v_x\ v_y\ v_z]$ in m/sec, (a) [0.5 0.5 1 0.5 0.25 -2], (b) [0.5 0.5 1 0.25 0.125 -1].

The relation between translational velocity vectors and tMMBM will not be given at this moment since the dependency on the scene depth is a dominant variable of the tMMBM. But the analysis of scene depth will be covered in detail in the next chapter where the effects of all parameters will be analyzed. Although, tMMBM refers to the metric that also involves translational motion of the camera, in the following sections tMMBM will be used to identify both MMBM and tMMBM since the major contribution will be shown to arise from MMBM and adding the translational velocity measurements unnecessarily complicate the experiments.

CHAPTER 4

ANALYSIS OF THE (MMBM) MEASURE

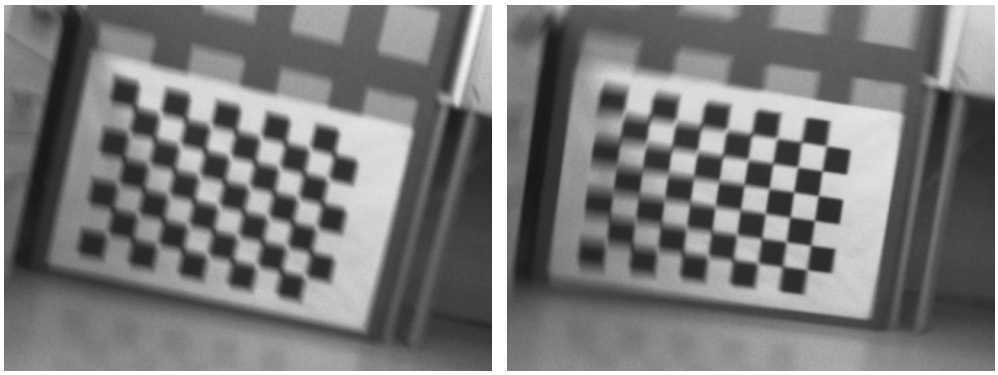
MMBM is a metric that gives a quantitative value for the amount of motion blur that will occur on an image due to egomotion of camera. There are three problems that needs to be addressed; (i) Does the derivations of MMBM really give information about motion blur?, (ii) How can MMBM be calculated in real time? and (iii) How does MMBM is affected from measurement noise. Following sections will answer all of the questions respectively.

4.1 MMBM VALIDATION

MMBM is based on the calculation of the optical flow vectors using inertial measurements instead of using two consecutive images. The assumption of constant velocity motion from the inertial measurement instant to the end of the exposure time make the optical flow comparable to the actual motion blur. Under the assumption of constant camera rotational velocity, MMBM is proportional to the average of optical flow vector magnitudes. The whole theory of MMBM that is previously explained in the previous chapter is based on the relation of optical flow and motion blur. Hence, MMBM should give consistent results with conventional optical flow calculation algorithms found in the literature. The relation and differences of optical flow and MMBM will be examined in this section.

In order to validate the metric, a hardware setup consisting of a camera, a 3D fiber-optic gyro and a PC was built to collect datasets. The translational motion of camera is considered to be negligible in this case. Camera and gyro axes were matched

with a fixture, but no calibration was done to obtain further alignment information. Using the handheld hardware, time synchronized image frames and gyro data samples were collected in the PC. Then, the method proposed by [38] was used to calculate the optical flow between all successive frames on the collected images. Two of the successive images from the data set are shown in Fig. 4.1.



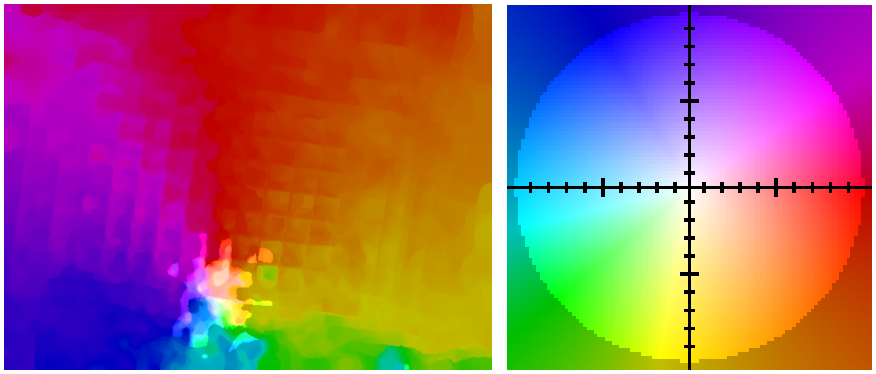
(a) (b)

Figure 4.1: Two successive images from the sequence recorded to validate the MMBM.

The result of an optical flow calculation for two consecutive images is a vector field that shows how much each pixel moved from the first image to the second one. For example, color coded optic flow field between Fig. 4.1(a) and Fig. 4.1(b) is shown in Fig. 4.2(b). Direction and intensity of optic flow vectors can be visualized with the help of the color map given in Fig. 4.2(a). Colors in the map are mapped to polar angles which optical flow vectors point through. For instance, red means a pixel is moving through east and yellow means the pixel moves through south. Moreover, intensity of each color gives normalized magnitude of movement in optical flow field.

In order to compare MMBM with optical flow the algorithm, average magnitude of optic flow (AMOF) vectors are calculated for each consecutive image frame. In other words, the magnitude of optical flow vectors for each pixel is averaged for the image sensor. AMOFs are then compared with MMBM values calculated from gyro data only. Optic flow algorithms assume that the exposure times of input frames are infinitesimal. So, instead of using the trigger instant, mid-exposure time is used to time-stamp each input image. The resulting output actually shows the displacement of each pixel. Although, MMBM convert instantaneous velocities of camera to ve-

locities of moving pixels, under the assumption of constant velocity during exposure time of camera, AMOF is expected to be comparable to MMBM. 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 of two waveforms since both give an idea about the magnitude of motion blur under aforementioned assumptions.



(a) (b)

Figure 4.2: Optic flow field from images in Fig. 4.1: (a) The actual optic flow field, (b) optic flow color and direction map.

Comparison of MMBM and AMOF is shown with two different data sets. 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. The results for the first data set is illustrated in Fig. 4.3. The camera motion is this data set mainly consists of yaw rotation. The plot 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 does not. Red dots on the AMOF plot correspond to mid points of time durations that are previously explaines in this section. AMOF data can only be calculated on the red dots, with the rest dashed lined being a piecewise linear interpolation in between.

On the second data set, the camera was subjected to a relatively faster and complex

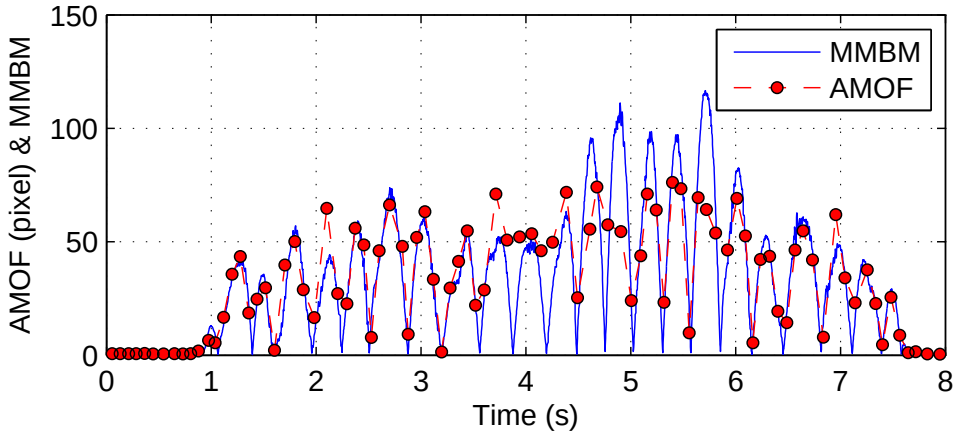


Figure 4.3: Comparison of scaled MMBM and AMOF during yaw axis camera rotations.

rotational motion with respect to each axis changed continuously and arbitrarily in time. It can be observed that discrepancies in Fig. 4.4 are more pronounced. The main reason for this is the frequency of rotations are faster than the rotation given in Fig. 4.4. Camera data starts to suffer from low sampling rate and the resulting aliasing. Moreover, optical flow algorithms are expected to be used on sharp images and their performance is 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.

4.2 REAL TIME MMBM CALCULATION

MMBM was evaluated numerically in Section 4.1 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 (3.10) 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. 4.5. The final MMBM calculation hence reduces

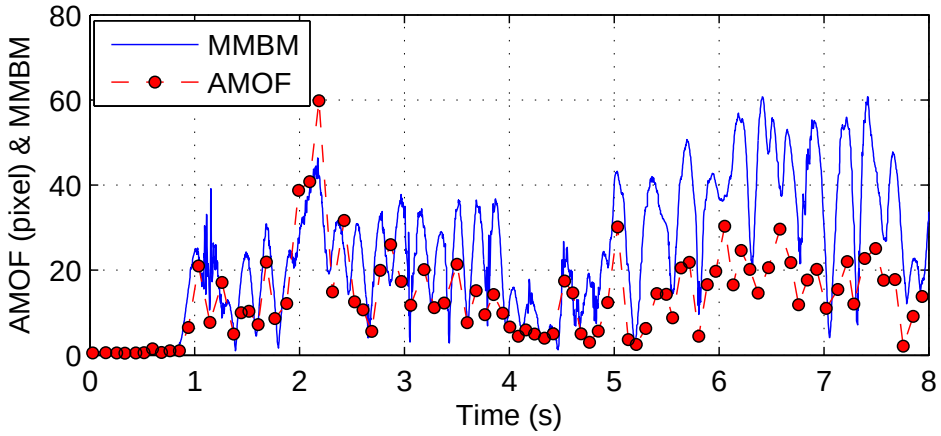


Figure 4.4: Comparison of scaled MMBM and AMOF during arbitrary and relatively faster camera rotations.

to

$$\mu^* = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\dot{u_i}^2 + \dot{v_i}^2}dA. \tag{4.1}$$

where $\dot{u}^2+\dot{v}^2$ is given by Eq. (3.17) 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 (4.1) 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. 4.6, with the small percentage errors justifying our choice of the number of samples used for the approximation.

4.3 SENSITIVITY ANALYSIS

One of the fundamental questions that must be answered is what happens if the measurements are noisy or erroneous. In real life inertial sensors, especially commercially

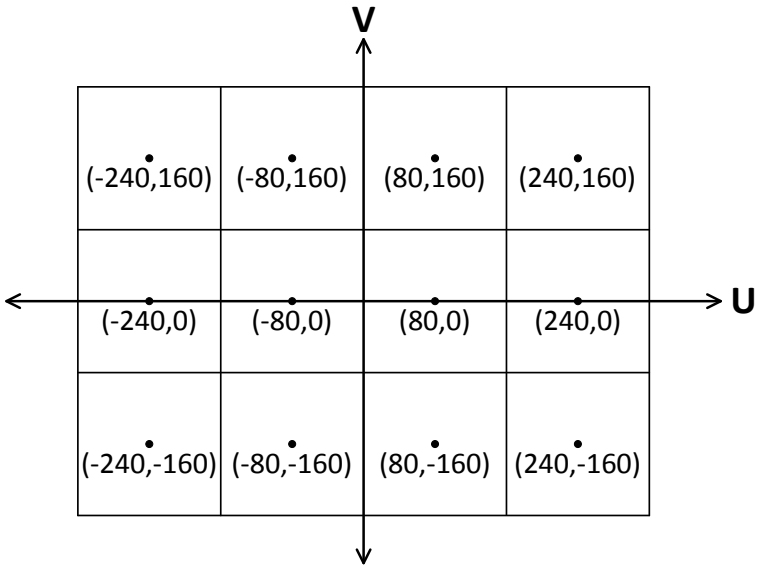


Figure 4.5: Areas (dA) and value evaluation points used for Riemann sum approximation.

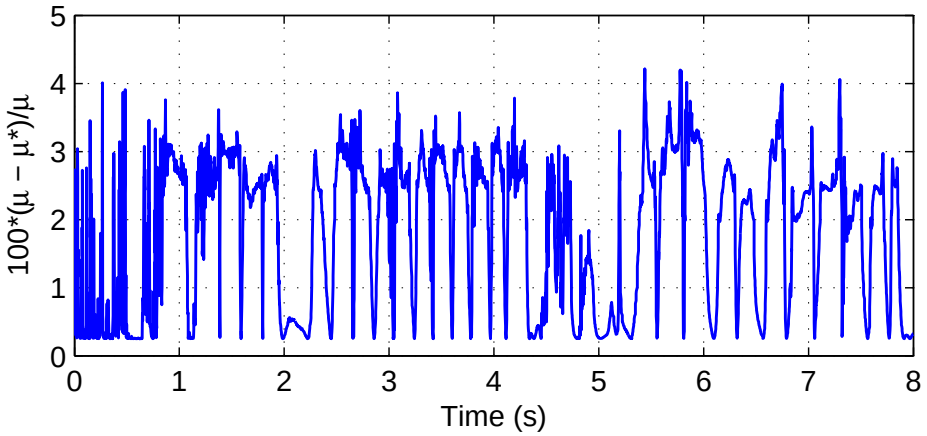


Figure 4.6: Percentage error between the numeric evaluation (μ) and the Riemann sum approximation of MMBM (μ*).

available cheap MEMS based ones are notorious for their noise levels. In this section the translational motion is also included in analyses, since all of the measurements will be predetermined values for demonstration purposes. Variation of MMBM under the noisy conditions will be examined in the rest of this chapter.

4.3.1 Sensitivity of MMBM to Focal Length "f"

Intrinsic calibration of camera showed that the focal length, f, is 4.2548mm. It is a very small focal length since we were using a wide angle lens. Like in all of the measurements, camera calibration can have noise or it may be calculated completely wrong. Therefore, knowing how MMBM changes for different focal lengths is crucial. Table 4.1 shows the MMBM value for noisy and non-noisy velocity measurements. Also the percentage of change is added to compare rate of changes in MMBM and f. Translation and rotation in z axis is not affected. That is also obvious since f has no relation with ω_z and v_z as seen in (3.20). But, both rotation and translation in x and y axis related motion blur is almost directly proportional to f variation, but the rate of changes are not exactly the same for MMBM and the f. Hence, the value of f should be calculated correctly to have correct camera egomotion to motion blur relation. Otherwise, effect of z axis motion may be overestimated or underestimated in MMBM calculation. If MMBM would proportionally change with f, it would still be used in external camera triggering application that will be explained in chapter 5 since the relative values of MMBM is the key point in that application.

4.3.2 Sensitivity of MMBM to Noise in Pitch, "w_x", and Yaw, "w_y" Motion

Pitch and yaw axis rotation velocities are actually directly proportional to MMBM as seen in the first four row of Table 4.2. However, motion components in all axes are added to each other and the final MMBM is the result of all added vectors. Vectors caused by different axis motion either constructively or destructively added to each other depending on the direction they are pointing through. Hence the value of MMBM can increase or decrease as the last two rows of Table 4.2 indicates. It can be hard to visualize combined motion blur vectors. However, in the worst case MMBM

Table 4.1: Effect of focal length parameter on MMBM for different velocity vectors

$[\omega_x, \omega_y, \omega_z, v_x, v_y, v_z, z]$	**f Noise**	**MMBM**	**MMBM +** σ	$\%$ **Change**
[1,0,0,0,0,0,0.5]	5%	4.3514	4.5594	4.7795%
[1,0,0,0,0,0,0.5]	-10%	4.3514	3.9372	-9.5204%
[0,0,1,0,0,0,0.5]	5%	0.9951	0.9951	0%
[0,0,1,0,0,0,0.5]	-10%	0.9951	0.9951	0%
[1,0,1,0,0,0,0.5]	5%	4.4020	4.6073	4.6642%
[1,0,1,0,0,0,0.5]	-10%	4.4020	3.9940	-9.2674%
[1,1,1,0,0,0,0.5]	5%	6.2514	6.5412	4.6346%
[1,1,1,0,0,0,0.5]	-10%	6.2514	5.6757	-9.2099%
[0,0,0,1,0,0,0.5]	5%	8.5095	8.9350	5%
[0,0,0,1,0,0,0.5]	-10%	8.5095	7.6586	-10%
[0,0,0,0,0,1,0.5]	5%	1.9902	1.9902	0%
[0,0,0,0,0,1,0.5]	-10%	1.9902	1.9902	0%
[0,0,0,1,0,1,0.5]	5%	8.6073	9.0279	4.8860%
[0,0,0,1,0,1,0.5]	-10%	8.6073	7.7680	-9.7509%
[0,0,0,1,1,1,0.5]	5%	12.1280	12.7253	4.9246%
[0,0,0,1,1,1,0.5]	-10%	12.1280	10.9350	-9.8370%

would change directly proportional with ω_x and ω_y variations.

4.3.3 Sensitivity of MMBM to Noise in Roll, "w_z" Motion

Similar comments said in Section 4.3.2 are also valid for the variations in roll axis rotation too. However, Table 4.3 also confirms that roll axis rotation is much recessive compared to pitch and yaw axis rotations. MMBM would change proportinally with changing z axis rotation variations only if the motion was purely z-axis rotation. But, when motion has other rotational components, the effects of noise in z axis rotation becomes significantly less important.

4.3.4 Dependence of MMBM to Scene Depth, "z"

The reason why only the rotational motion was considered in real experiments will be clear in this section. When scene depth increases, the motion blur resulting from the same motion significantly shrinks. Especially field robots like SensoRHex usually

Table 4.2: Effect of noise in pitch and yaw parameter on MMBM for different velocity vectors

[$\omega_x, \omega_y, \omega_z, v_x, v_y, v_z, z$]	**Noise**	**MMBM**	**MMBM +** σ	$\%$ **Change**
[1,0,0,0,0,0,0.5]	5$\%$ ω_x	4.3514	4.5690	5$\%$
[1,0,0,0,0,0,0.5]	-10$\%$ ω_x	4.3514	3.9163	-10$\%$
[0,1,0,0,0,0,0.5]	5$\%$ ω_y	4.4273	4.6486	5$\%$
[0,1,0,0,0,0,0.5]	-10$\%$ ω_y	4.4273	3.9846	-10$\%$
[1,0,1,0,0,0,0.5]	5$\%$ ω_x	4.4020	4.6170	4.8852$\%$
[1,0,1,0,0,0,0.5]	-10$\%$ ω_x	4.4020	3.9728	-9.7496$\%$
[1,1,0,0,0,0,0.5]	5$\%$ ω_x & ω_y	6.2063	6.5166	5$\%$
[1,1,0,0,0,0,0.5]	-10$\%$ ω_x & ω_y	6.2063	5.5857	-10$\%$
[1,1,1,0,0,0,0.5]	5$\%$ ω_x	6.2514	6.4041	2.4423$\%$
[1,1,1,0,0,0,0.5]	-10$\%$ ω_x	6.2514	5.9583	-4.6886$\%$
[1,1,0,1,0,0,0.5]	5$\%$ ω_x	13.6473	13.7181	0.5193$\%$
[1,1,0,1,0,0,0.5]	-10$\%$ ω_x	13.6473	13.5149	-0.9697$\%$
[1,1,0,0,0,1,0.5]	5$\%$ ω_x	6.3850	6.5375	2.3871$\%$
[1,1,0,0,0,1,0.5]	-10$\%$ ω_x	6.3850	6.0917	-4.5939$\%$
[1,1,0,1,0,1,0.5]	5$\%$ ω_x	13.7110	13.7820	0.5179$\%$
[1,1,0,1,0,1,0.5]	-10$\%$ ω_x	13.7110	13.5783	-0.9674$\%$
[1,1,0,1,1,1,0.5]	5$\%$ ω_x	13.6512	13.5861	-0.4768$\%$
[1,1,0,1,1,1,0.5]	-10$\%$ ω_x	13.6512	13.7908	1.0227$\%$

work in large fields and the scene depth is usually more than a few meters. As seen in Table 4.4 motion blur caused by the translational motion can be quite large for close scenes and it becomes very small once scene depth goes beyond a few meters.

Percentage of rotational and translational motion effects can significantly change for different scene depths. Table 4.5 illustrates MMBM values calculated for only rotation measurements and for whole motion of camera for different scene depths. The given motion is a reasonable upper bounds for uniform slow locomotion of SensoRHex and it is the same throughout Table 4.5. The only control variable is the scene depth. Effect of translation can be dominant for close camera to scene distance, but, it is negligibly small for cameras working on large fields, in particular when the scene depth is more than a few meters.

Table 4.3: Effect of noise in roll parameter on MMBM for different velocity vectors

$[\omega_x, \omega_y, \omega_z, v_x, v_y, v_z, z]$	**Noise in z**	**MMBM**	**MMBM + σ**	**% Change**
[0,0,1,0,0,0,0.5]	5%	0.9951	1.0449	5%
[0,0,1,0,0,0,0.5]	-10%	0.9951	0.8956	-10%
[0,1,1,0,0,0,0.5]	5%	4.5104	4.5190	0.1891%
[0,1,1,0,0,0,0.5]	-10%	4.5104	4.4946	-0.3506%
[1,0,1,0,0,0,0.5]	5%	4.4020	4.4073	0.1212%
[1,0,1,0,0,0,0.5]	-10%	4.4020	4.3922	-0.2229%
[1,1,1,0,0,0,0.5]	5%	6.2514	6.2561	0.0740%
[1,1,1,0,0,0,0.5]	-10%	6.2514	6.2429	-0.1371%
[0,0,1,1,0,0,0.5]	5%	8.5522	8.5566	0.0513%
[0,0,1,1,0,0,0.5]	-10%	8.5522	8.5441	-0.0950%
[0,0,1,0,0,1,0.5]	5%	2.2252	2.2479	1.0198%
[0,0,1,0,0,1,0.5]	-10%	2.2252	2.1825	-1.9184%
[0,0,1,1,0,1,0.5]	5%	8.6486	8.6529	0.0492%
[0,0,1,1,0,1,0.5]	-10%	8.6486	8.6408	-0.0910%
[0,0,1,1,1,1,0.5]	5%	12.1786	12.1824	0.0311%
[0,0,1,1,1,1,0.5]	-10%	12.1786	12.1714	-0.0592%

Table 4.4: Effect of scene depth on MMBM

$[\omega_x, \omega_y, \omega_z, v_x, v_y, v_z]$	**z**	**MMBM**
[0, 0, 0, 0.25, 0.125, -1]	0.2	7.1647
[0, 0, 0, 0.25, 0.125, -1]	1	1.4329
[0, 0, 0, 0.25, 0.125, -1]	5	0.2866
[0, 0, 0, 0.25, 0.125, -1]	20	0.0716

Table 4.5: Contribution of translation on MMBM depending on scene depth

$[\omega_x, \omega_y, \omega_z, v_x, v_y, v_z]$	**z**	**MMBM**	**tMMBM**	**Tr. Contribution**
[0.5, 0.5, 1, 0.25, 0.125, -1]	0.2	3.1935	8.4038	163%
[0.5, 0.5, 1, 0.25, 0.125, -1]	1	3.1935	3.8547	20.7%
[0.5, 0.5, 1, 0.25, 0.125, -1]	5	3.1935	3.2931	3.1%
[0.5, 0.5, 1, 0.25, 0.125, -1]	20	3.1935	3.2163	0.71%

CHAPTER 5

MMBM FOR MOTION BLUR MINIMIZATION

The definition, derivation and analysis of MMBM were given in previous chapter and the different possible areas and applications where MMBM can be used will be explained in the current chapter. The most prominent ability of MMBM is tracking the motion blur in real time. Having a notion of motion blur level is a very valuable information for the user. This information can be very crucial and leads ways too many applications. Definitely, the image processing approaches are valuable for estimating motion blur caused by movement of individual objects in the scene. But, the limitations of such approaches are motion blur calculation frequency (approx. 30Hz) and the fact that a blurry image must be captured for the estimation. The calculation of MMBM can be done at very high frequencies (more than 600Hz) compared to image processing based motion blur estimation that enables the user to track motion blur caused by quickly changing camera motion. Furthermore, using a complementary sensor to camera gives flexibility to do some processes before image is captured.

In order to understand possible applications, the image acquisition by using a complementary inertial sensor must be understood. Fig. 5.1 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_{im}. 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 compu-

tation 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.

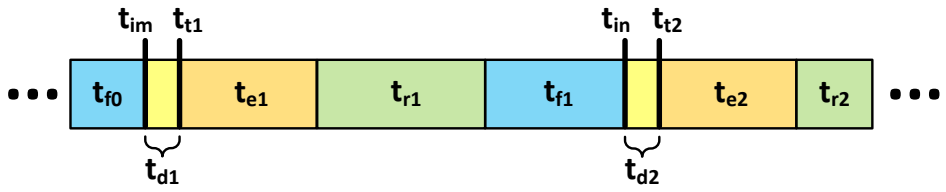


Figure 5.1: Timing diagram of capturing an image and using an inertial sensor.

MMBM can be directly used in motion blur related scenarios. Giving decisions on motion blur level of an image and manipulation of camera triggering instances to obtain sharper images are directly available thanks to the information MMBM gives. It is also possible to improve performance of MMBM with some additions for the actual image capture process. All of the details related to the different applications of MMBM will be discussed in the following sections.

5.1 MMBM FOR FRAME TRIGGERING BASED MOTION BLUR MINIMIZATION

An intuitive approach to triggering an image frame is measuring rotational velocities of a robot with a gyroscope at a high sampling frequency and triggering the camera when the magnitude of the angular velocity vector is sufficiently small. Although, this idea has potential, rotational velocities around the three different axis do not contribute to motion blur equally. However, the proposed metric models the effects of all axes rotations on motion blur. Individual rotational velocities and corresponding metric values while SensoRHex is walking straight on a flat surface can be seen in Fig. 3.4. The plot spans approximately 2.5 step cycles of the hexapod. Naturally, exact motion blur depends on MMBM levels during the entire camera integration time. However, triggering an image when MMBM is sufficiently low is expected to give sharper images since the acceleration of a robot with considerable inertia (10 kg

mass for our platform), would be limited.

Triggering a camera externally involves reading gyro data and calculating MMBM in real time and externally triggering the camera according to MMBM value. The timing diagram given in Fig. 5.1 can be used to explain externally triggering a camera with calculated metric value. Gyro data can be collected at a few hundred Hz and for each gyro data MMBM can evaluated in real time. When camera is free, e.g. t_{f0}, and MMBM is sufficiently low for gyro data sample obtained at t_{im}, external trigger pulse is applied to the camera at t_{t1}. There is a delay t_{d1}, which is approximately 2-3ms, between gyro data sample and trigger moment since MMBM is evaluated in a PC and trigger command is passed to a microcontroller which generates the trigger pulse. As soon as the trigger pulse is generated exposure time t_{e1} begins and elapses approximately 10-50ms. Most of the cameras can perform image acquisition and sending a frame data to a PC in parallel. However, we preferred to work on serial mode in which user has to wait for data to be transferred, t_{r1}, before starting a new exposure. Because, triggering instance in parallel mode cannot be applied as precisely as the external triggering in series mode. Finally, camera becomes available for capturing a new image frame at t_{f1}.

Camera triggering with MMBM relies on the idea that only sharp images should be captured. Once the image is blurred, reverse filtering approaches can be applied. However, deconvolution is an ill defined problem and deblurring is usually done with search methods which require considerable computational power and most of the deblurring algorithms cannot be applied in real time applications.

The aforementioned triggering strategy was implemented on SensoRHex. The camera is triggered only if it is free, the last three samples of MMBM is lower than a constant threshold and those last samples are monotonically decreasing. In order to fairly explain the difference of uniform and MMBM triggering, maximum camera sampling is assumed to be 5 fps and triggering of a new frame was allowed only after 0.2 seconds passed since the last triggering instance and all triggering conditions are met. Therefore, it is guaranteed that number of images captured by MMBM triggering will be lower than what what camera can capture at its maximum speed. The resulting triggering scheme can be seen in Fig. 5.2. Although, triggering signal is given when

the metric value is substantially low, it can get larger during an exposure period. Hence, the resulting image may become moderately corrupted by the motion blur. But, the extreme blur cases are completely avoided. Triggering the camera at time instances when the exposure time overlaps with a minima of the metric also requires predictive techniques to optimally minimize motion blur.

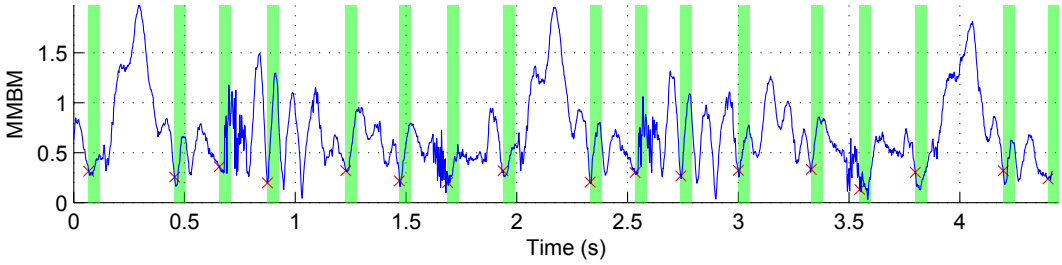


Figure 5.2: MMBM based and image capture. Shaded regions illustrate exposure time and crosses pinpoint the trigger instants.

Using the triggering approach explained in this section, extremely blurred images, which would have too much information loss, can be avoided and sharp image throughput can be maximized. Practically, this application is one of the most appealing one because of maximization of the number of sharp images and its simplicity. The detailed performance of this approach is given in chapter Section 6.

5.2 MMBM WITH PREDICTIVE FRAME TRIGGERING

The actual motion blur is a function of camera movement and/or object movement on the scene during exposure period. One idea is to combine two techniques mentioned in Section 5.3. More clearly, the application can be triggering the camera externally with the average MMBM calculated during the exposure period. This technique is non-causal and requires the knowledge of how the camera will move in the future. Although, the implementation of that method sounds like impossible, there exists forecasting methods for the systems whose behaviour can be modelled and is predictable.

There are two possible scenarios where this approach can be used. First, the camera motion may be controlled by the user, so that, the future motion is actually known.

For instance, when a camera is mounted on a serial manipulator arm and the motion of the controller of the arm applies predefined trajectories for the end effector. Thus, the exact future motion of the camera is known and the future values of MMBM can be calculated by using that knowledge instead of obtaining the motion information by an external sensor that limits the application to the causal domain. Having an average MMBM value for the whole exposure period can be used to capture images such that exposure period will reside on a local minima of MMBM. Thus, the quality of captured images would be optimal. In the second scenario, the motion of the camera is not predetermined, however, the body that is carrying the camera has a motion model. Motion models give predicted motion of a body considering the current states and the inputs. For instance, a motion model of a car can predict what will be the position and velocity of the car, if you throttle the gas pedal and steer the wheels for certain amounts. Inevitably, models cannot incorporate all of the variables that would affect the motion, but, only considers the most dominant effects. Better the derived model, better the predictions will be. Whenever prediction is possible, MMBM can be calculated for the estimated future motion. Hence the camera can be triggered accordingly.

In most of the robotic areas, the robot motion is not predetermined and the motion model of the robot may not exist. The robot that is used for the validation is a six legged robot whose legs are c-shaped and compliant. Although, the flexibility of legs is a very crucial factor on the stability of the robot, due to the same flexibility SensoRHex does not have any motion model. But, it may still be possible to predict the near future robot motion by using blind forecasting methods, if the robot body is exhibiting almost periodic motion for certain locomotion type. Fig. 5.6(a) and Fig. 5.2 gives almost 5 steps of robot and the calculated MMBM values seems to be a periodic function with certain amount of noise to human perception. There is an open door for improving the performance of MMBM based camera triggering. In this thesis, no prediction algorithms are incorporated. However, we tried to analyze how much improvement would be possible, if the near future motion of SensoRHex could have been perfectly predicted.

Saving images and MMBM values calculated online to memory gives opportunity to do offline analyses on them. In our case, we did not implement any prediction al-

gorithms, however, we know what would be the MMBM values for a fixed exposure time for all of the possible trigger instances since we saved all of the data. Fig. 5.3 shows the regular MMBM in blue plot and average value of all MMBM samples for the next 50ms in red. Red line is smoother than the blue one since it averages multiple MMBM values over 50ms and there exists a time delay between them. Although the horizontal displacement of curves is not very high, actual values, vertical displacement, can significantly change. The difference between MMBM and the perfect estimation of averaged MMBM is shown is Fig. 5.4. Acceptable blur amount threshold was set to approximately 0.4. Even though camera was triggered when MMBM was below the predetermined threshold, variations in the velocity of camera can change the actual motion blur amount almost 0.7 units higher than the instantaneously calculated MMBM. This fact results in more blurry images than the expected ones. The percentage difference can be as high as 150% as seen in Fig. 5.5. More quantitative analysis can be found in Table 5.1. As expected, the mean error is close to zero since the perfectly estimated MMBM can be higher or lower than the instantaneously estimated MMBM. But, the mean of absolute value of error turns out to be 0.1930 and the maximum difference is 0.7465. Those numbers are not negligible when the maximum value of MMBM is considered to be approximately 2.0. Percentage wise, improvements up to 150% and on the average 30.14% seems to be possible using a prediction scheme assuming ideal prediction is possible. The reason why the improvement percentage can be very high is we are already capturing sharp images that correspond to low MMBM values. Hence, having a 0.75 unit error can really correspond to very high percentages. But still, 30% average improvement would be a good contribution if we could have predicted the future motion of SensoRHex perfectly. Unfortunately, RHex does not have any motion model and only alternatives are blind forecasting techniques for which, the improvement percentage is expected to be less than 30%.

Table 5.1: Potential performance gain from prediction.

Parameter	**Min.**	**Max.**	**Mean**	**Abs. Mean**	**Std. Dev.**
Error	-0.7465	0.7397	0.0028	0.1930	0.2448
% Error	-94.47%	147.84%	3.88%	30.14%	37.92%

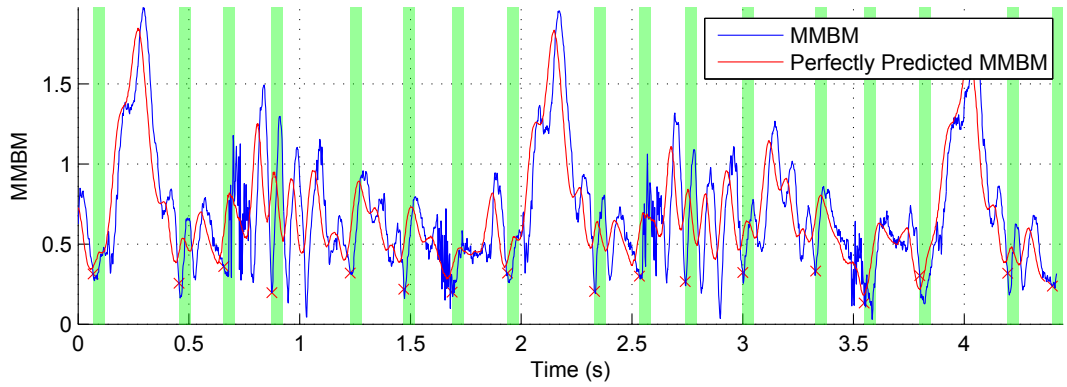


Figure 5.3: Comparison of MMBM and the average of perfectly estimated MMBM for the next 50ms.

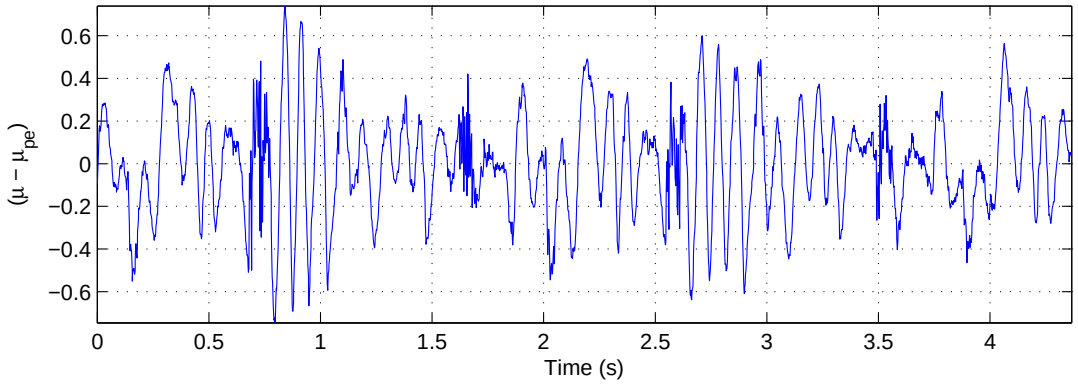


Figure 5.4: Error between MMBM and the average of perfectly estimated MMBM for the next 50ms.

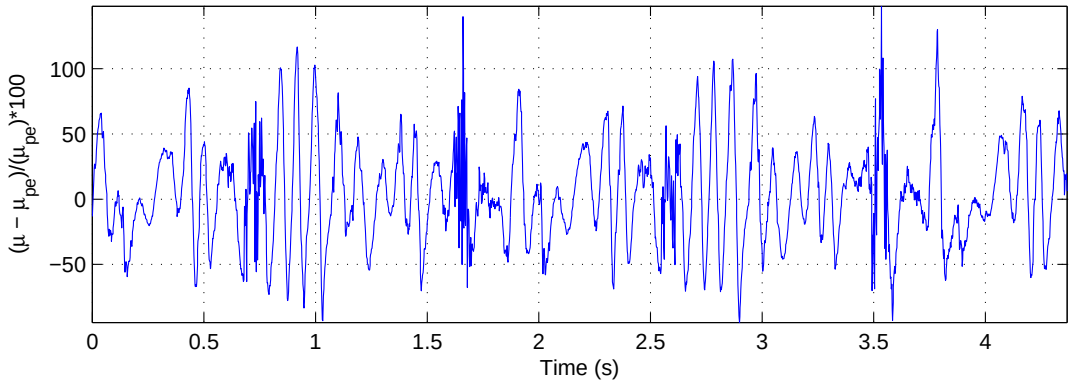


Figure 5.5: Percentage error between MMBM and the average of perfectly estimated MMBM for the next 50ms.

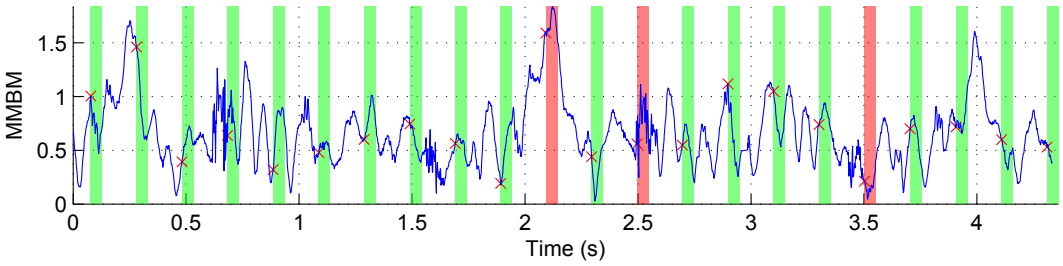
5.3 OTHER POTENTIAL USES FOR MMBM

MMBM gives an estimation of motion blur when the camera velocity is assumed to be constant during the exposure period. It is just an instantaneous information that is calculated from each gyroscope data sample obtained. Higher values of the metric corresponds to higher amount of average motion blur and lower values of it represents sharper images if an image capture begins at that moment.

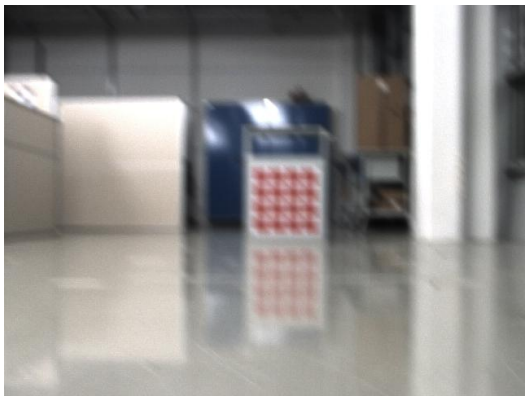
Fig. 5.6(a) illustrates the MMBM value variation while SensoRHex is walking straight on a flat surface. Plot spans slightly larger time duration than two steps of SensoRhex. Camera was triggered at uniform 5fps during the locomotion and exposure time of the camera was set to 50ms. The red crosses on MMBM plot marks the trigger instances and shaded regions illustrate exposure periods. Trigger instances are uniformly separated since the frame rate is fixed. Exposure periods reside on random locations on the metric plot since triggering and locomotion were two independent actions. When the exposure period overlapped with a maxima of MMBM, marked as the first red shaded region on Fig. 5.6(a), the image in Fig. 5.6(b) was captured. In that particular image motion blur is so extreme that red blobs on the 4x4 square blob pattern cannot be independently identified. When the exposure period resides on moderate levels of MMBM as seen on the second red shaded region on Fig. 5.6(a), blobs can be separately identified, even though the image still has motion blur. Moreover, if an image is triggered such that exposure period coincides with a local minima of MMBM, marked as the last red shaded region in Fig. 5.6(a), a pretty sharp image in Fig. 5.6(d) is captured. Similarly, if the MMBM remains in the moderate levels during the exposure time, the resulting images are moderately blurred. Therefore, triggering images when MMBM is low and remains low during exposure period results in less blurred images.

5.3.1 Frame Quality Assessment Using MMBM

The most simple usage of MMBM can be on giving decisions whether or not to use an image. One embodiment can be using the camera on the regular fixed frame rate capturing mode and giving a decision on whether to directly use each image



(a)



(b)

(c)

(d)

Figure 5.6: Relation between image sharpness and MMBM. Shaded regions illustrate exposure time and crosses pinpoint the trigger instants. Onboard camera is (a) uniformly sampled at 5fps. Example frames given in (b), (c) and (d) are those triggered consecutively at red shaded regions in (a).

depending on a single MMBM value calculated at the beginning of camera's light exposure. Fig. 5.6(a) illustrates a convenient example for this usage. Camera works at fixed 5 fps rate and MMBM is calculated independently using rotational velocities of camera. One can simply discard an image if the motion blur level of the image is known to be too high that no meaningful information can be obtained from that image. For instance, image shown in Fig. 5.6(b) can be discarded as soon as it is captured, if that amount of motion blur makes the image useless by just comparing the MMBM value calculated at the beginning of that image's exposure time with a predetermined threshold value. Such a usage can be useful in cases where losing some of the images are not so crucial and user can tolerate waiting for the next image. Although, some of the images would be discarded, user will be working with images that are meaningful to him. Instead of wasting computation time to extract some features from images and still having no information at all or a divergent information, CPU can be used for some other purposes while waiting for the next image. An enhancement to this approach can be taking the average of calculated MMBM values for a whole exposure period, i.e. t_{e1}. User will have both image and MMBM values at the end of the exposure period. Then, using the average of MMBM values is expected to give better sharp/blurred image classification rate.

The MMBM can always be calculated while the system is on, i.e. during t_{f0}, t_{d1}, t_{e1}, t_{r1}, t_{f1} and so on. Motion blur information that comes from MMBM is calculated at high sampling rate compared to image acquisition and it can be calculated multiple times even for one exposure period. Therefore, it can be possible to decide not to use an image right before or during the capture process, if the hardware of camera is designed accordingly. In this way, user would not even capture any images and would not need to save it to the memory.

Discarding an image is not the only decision that can be made. Any decision that requires the motion blur amount on a captured image can be made by using MMBM. For example, there exists image sharpening methods to remove the motion blur up to a certain extend. These methods are usually time consuming and requires considerable amount of computation power due to the ill-defined nature of the problem. The decision of whether or not to use sharpening methods can be made depending on the MMBM. If the MMBM values remained high during the exposure period, the image

will be highly blurred as seen in Fig. 5.6(b) and a resharpening method can be applied to only this image while the images like Fig. 5.6(c) and Fig. 5.6(d) are directly used.

5.3.2 Blur Kernel (PSF) Estimation

The aim of MMBM is to avoid motion blur, before it actually occurs. Inevitably, there will be certain amount of motion blur on images unless the camera is standing still and the objective was delaying image acquisition until more favorable image capturing instances occur. Since the output images of MMBM based camera triggering are sharper than fixed frame rate recording, applying resharpening algorithms to those images are expected give even better results. Furthermore, resharpening algorithms also suffer from extreme motion blur since some of the information is completely lost to the blur and that information can not be reversible. Hence, avoiding extremely blurred frames works for resharpening algorithms too.

Most of the resharpening algorithms require the image kernel (or PSF) that represents the motion blur. Convolution of those kernels with sharp images would give the blurred image and inverse filtering the blurred image is one of the highly researched topics as explained in chapter Section 2. In real life, those kernels are spatially varying since the shape of blur is not the same throughout images for most type of motion. Some of the researches use computer vision techniques to find PSF, and some others use external sensor information. Inertial sensors appear to be among the most popular ones in PSF estimation. MMBM already uses a gyroscope and optionally a accelerometer. The measurements obtained from inertial sensors can also be used on PSF estimation.

5.4 DISCUSSION ON (DEALING WITH) NON-UNIFORM SAMPLED VIDEO

Once the MMBM based camera triggering is used, the captured images are not uniformly sampled any more. This may not be a problem for an operator who is controlling the robot by using the images captured from an onboard camera. Although refreshing rate of images will be different, operator will be seeing sharper images. However, this may still be a problem for the operator or different algorithms that

need information extracted from captured images.

5.4.1 Frame Interpolation

In cases where the uniformity of measurements is important, reconstruction of uniformly sampled video stream can be possible. Saving time stamps of images and gyro data, the movement of pixels can be estimated. Referring back to Fig. 5.1 and assuming t_{t1} is in the correct time instance and t_{t2} is slightly delayed, it is possible to extract the expected image if the fixed frame rate capture was used. In other words, interpolation based image reconstruction can be done by making use of both images and motion information obtained from inertial sensors. Finally, operator or the controlling algorithms would work with only uniformly reconstructed images. That kind of reconstruction needs further research on it to prove its applicability and it is not examined further in this thesis.

5.4.2 Feature Interpolation

Uniform reconstruction of whole video stream can be computationally demanding and/or unnecessary. Because, most of the measurements are taken using extracted features from images. In other words, features are more important than the whole image in most of the cases. MMBM based camera triggering is already shown to yield sharper images and features can be extracted more correctly from sharper images. Therefore, interpolating only the location of features is expected to save computation time, yet, give uniformly sampled measurements. Hence the conventional algorithms, which require fixed amount of time between each consecutive input, can be conveniently used with MMBM based camera triggering approach.

5.4.3 Bayesian Estimation / Filtering

There are algorithms which require information extracted from images and they can already handle non-uniform data. One of the areas where information extracted from images is heavily used is simultaneous localization and mapping (SLAM). There are

two common approaches used in SLAM: Kalman Filter and Particle Filter. It is possible to use non uniformly sampled measurements in both filters. They have basically two step which are the motion update and the measurement update. Motion update is done using the motion model of the robot and it gives a predicted location of robot. The uncertainty of SLAM increases with each motion update since the models incorporate noise to model real life more accurately. On the other hand, the measurement update gives new information and decreases the uncertainty, even though the measurements have specific noise too. Especially, particle filter is more suitable to work with non-uniform measurements. Simply, small, fixed and multiple motion update steps can be executed until a measurement arrives and the measurement update can be executed whenever the data is available. Similar approach can be applied in Kalman filter too. However, the analytical derivations would be more complex.

CHAPTER 6

PHYSICAL EXPERIMENTS

The advantages MMBM gives can best be expressed by showing it in action. A hardware setup is prepared to demonstrate how MMBM can be used. The hardware is mounted on SensoRHex to show image quality improvements. This chapter presents our results on the physical experimental setup and presents a comprehensive discussion of the results.

6.1 HARDWARE

The proposed MMBM based camera triggering system was implemented on the SensoRHex hexapod. The system consists of a Fizoptika optic gyro, a PointGrey Flea2 camera, a PIC based microcontroller board and a 500MHz Pentium class PC-104 CPU. Trigger decisions are made by the sensor PC, but, the camera is triggered with an interface microcontroller. Images are acquired with a PointGrey Flea2 camera. All hardware is mounted on SensoRHex, thus, the mobility of robot is assured. High level commands such as calibrate, sit, walk and run are given by a human operator via an external operator PC. Controller of SensoRHex actuates each leg depending on the commands given by the operator. On the robot side, main voltage and current measurements can be obtained from power node and hip nodes both actuate the motors and supply local data like motor voltage, motor current and motor encoder readings to the control PC of SensoRHex. All of the robot parameters are collected on the control PC and sent to the sensor PC over the ethernet connection. That data can be collected and stored or processed online depending on the needs of the user.

Furthermore, we have an external tracking system based on multiple cameras which gives accurate position of robots that can be used as ground truth on ATLAS. The robot pose is tracked with an external tracking system which has multiple cameras. Global position and orientation of the robot is sent to the sensor PC in real time. All of the sensor data is gathered in one PC which enabled us to give a time stamp. Although, slight delays occur during the data transfer, those delays do not exceed a few milliseconds and hence all data is conveniently synchronized within reasonable error range. Connections between all hardware components are illustrated in Fig. 6.1.

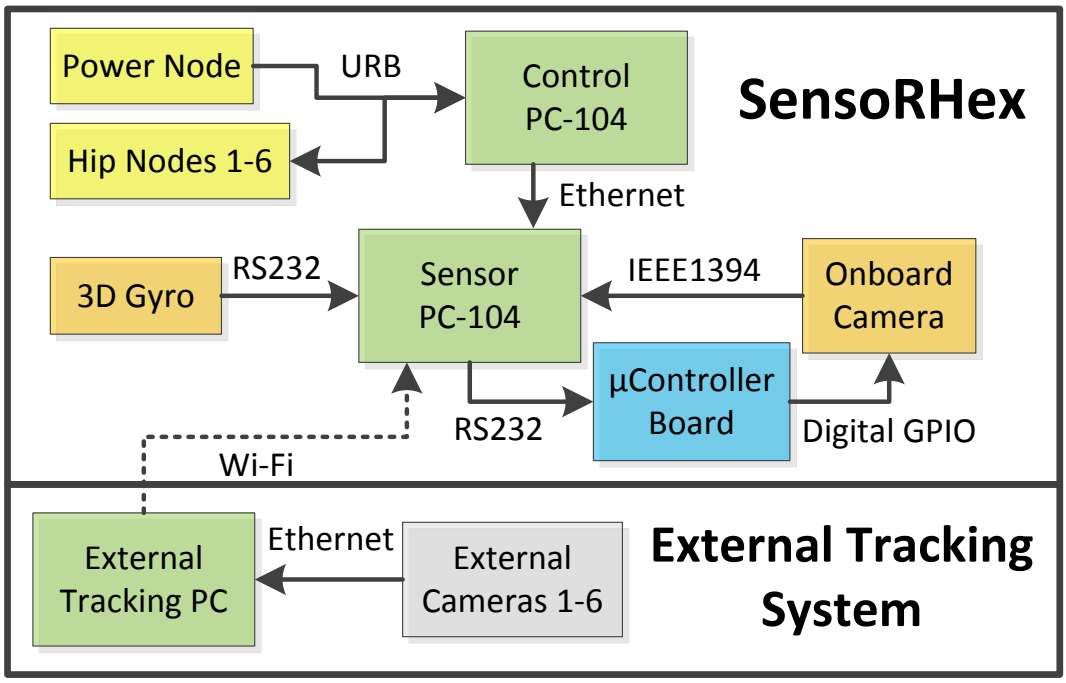


Figure 6.1: Experimental data collection setup.

All hardware components of MMBM based camera triggering system are located on SensoRHex. The gyro gives rotational velocities in each axis at 600Hz. Due to the simplicity and sufficiency concern only the rotational motion of the camera is considered throughout the experiments. MMBM is calculated for each gyro reading by a PC-104 CPU unit in real-time. Fig. 3.4 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. 3.4 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 is triggered only when MMBM is below a certain threshold and monotonically decreasing.

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

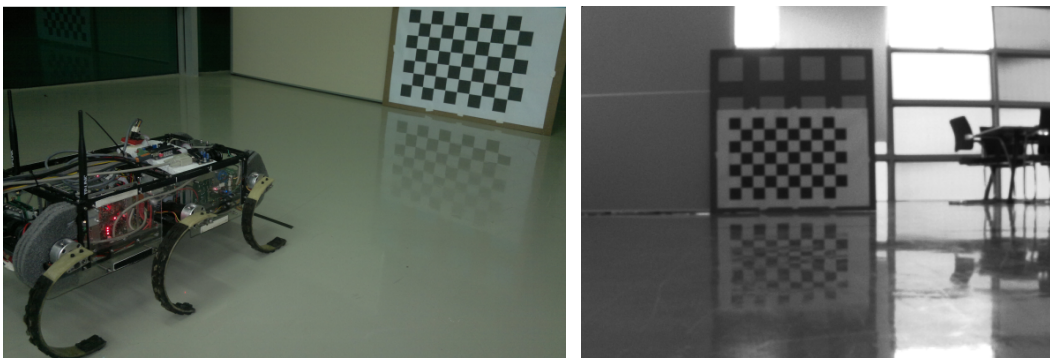
6.2 IMAGE SHARPNESS IMPROVEMENT WITH WALKING SENSORHEX

In this experiment the aim was to demonstrate image sharpness improvement when the MMBM base camera triggering is applied. 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. 6.2. 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 and monotonically decreasing.

Table 6.1: 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

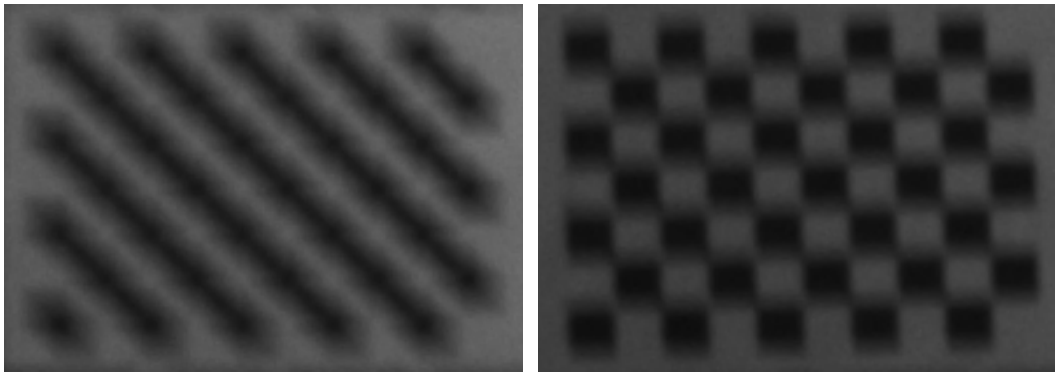
In order to quantitatively evaluate improvements on motion blur, checkerboard pat-



(a) (b)

Figure 6.2: Motion blur minimizing system implementation on SensoRHex. (a) Experiment area, (b) robot's point of view.

terns were cropped from each frame and cropped images were evaluated with Just Noticeable Blur Metric (JNBM) [8]. 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 6.1. 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. 6.3(a) and Fig. 6.3(b) show hand-picked extremely blurred frame examples from both data sets.



(a) (b)

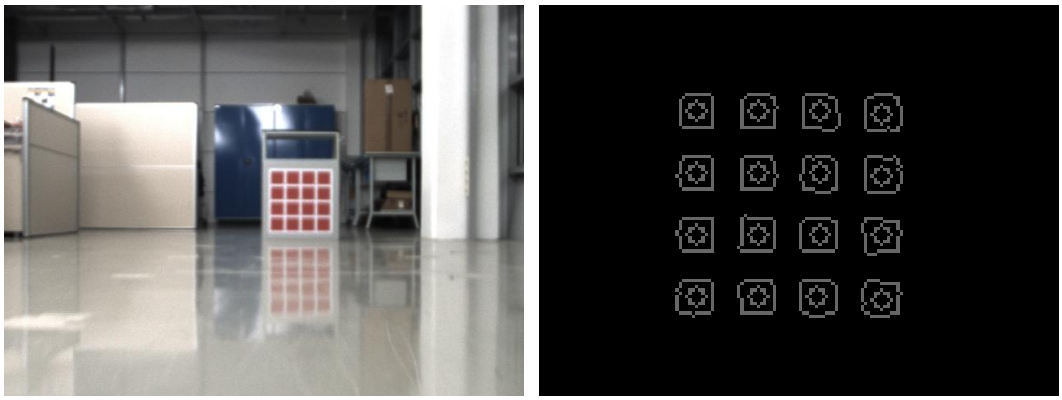
Figure 6.3: Examples of hand-picked excessively blurred frames from SensoRHex walking: (a) Image from fixed frame rate capture (b) Image from external triggering with MMBM.

6.3 IMPLICATIONS OF SHARPER IMAGES ON VISUAL PERCEPTION

Motion blur can cause serious information loss on images. The visual features computed on images are corrupted or may completely vanish. In order to assess the effectiveness of MMBM triggering, an experimental setup consisting of our experimental hexapod and an external motion tracking system was constructed. The main purpose of this experiment is to show motion blur corrupts features and how MMBM based camera triggering improves feature extraction performance.

SensoRHex walked straight through a similar pattern as illustrated in Fig. 6.4(a) at different constant walking velocities throughout the experiment. The optical target pattern in Fig. 6.4(a) has 16 identical 10cm by 10cm square red patches and all blobs have 2cm separation between each other, but, the pattern used in Table 6.2 was half of its size. Global positions of those patches' center points are known. The starting position of SensoRHex was 6 meters far away from the pattern and robot walked approximately three meters in each experiment. Robot walked at three different velocities. Fast, moderate and slow walks correspond to 0.53m/s, 0.17m/s and 0.085m/s respectively. While the robot was walking, images are captured with an onboard camera. Initially, uniform sampling at 5fps is used to get images. Then, MMBM triggering is applied at three different thresholds. For each metric threshold robot walked at all modes. For a fair comparison, only the data corresponding to the same walk duration is considered for each walk mode. All of the acquired images are color filtered to find blob positions on sensor plane. Once the world coordinates of red blobs and their corresponding image plane coordinates are known, camera location with respect to the world coordinates can be extracted, i.e. extrinsic camera calibration can be performed. However, the problem addressed in this experiment is extraction of blobs. Because, motion blur corrupts features, which are square red blobs in our case, and may result in extreme information loss as shown in Fig. 6.6(a). Captured images are processed offline. Color filter was applied to extract blobs. Fig. 6.4, Fig. 6.5 and Fig. 6.6 illustrates three different motion blur levels in a single run. Fig. 6.4(a), Fig. 6.5(a) and Fig. 6.6(a) are the raw images captured with the resolution 640x480 pixels. On the other hand, Fig. 6.4(b), Fig. 6.5(b) and Fig. 6.6(b) shows color filtered images. For better clarity, only the cropped regions of interests (160x120 pixels) are

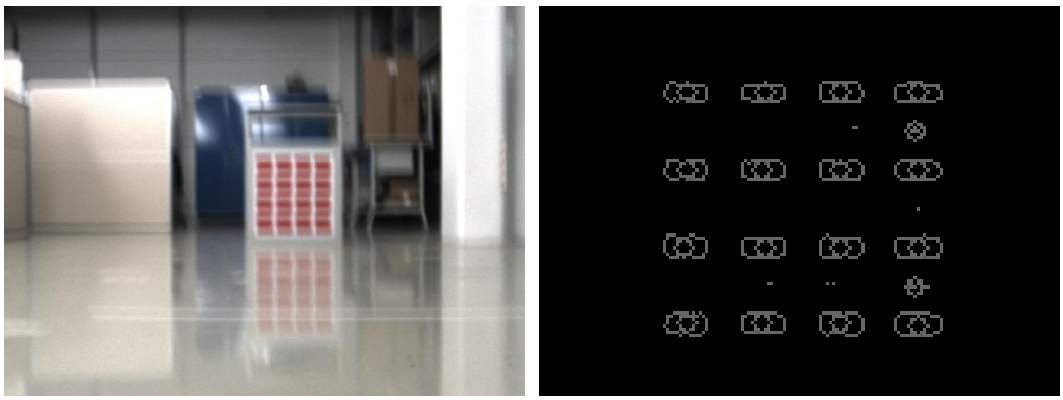
shown on the processed images. Due to motion blur, extract all 16 blobs could not been extracted in some of the images, for example, Fig. 6.6(b). Motion blur also causes distortions on images that can result in distorted features and false alarms as shown in Fig. 6.5(b) When some of the blobs are lost or many of them fused together, blobs cannot be matched with their global positions which makes the camera position estimation impossible. Feature extraction statistics are given in Table 6.2. Then, we applied extrinsic camera calibration using OpenCV libraries to find location and orientation of camera. Pose and orientation extracted from images and external tracking system are compared. We confirmed that once the blob center locations are extracted, location measurement performance from images are similar to each other. Although, distance in z-axis, which is distance between robot and the pattern, can be extracted with minimal error, orientation of camera is considerably noisy due to the planar pattern. Hence, the location in x-axis and y-axis suffers from the noise in orientation.



(a) (b)

Figure 6.4: Blob extraction from a sharp image. All blobs can be identified accurately. (a) The original image, (b) output of blob extraction.

The only parameter defined by the user to use the proposed triggering method is the threshold value. Threshold value states the maximum allowable motion blur amount in images. Setting the threshold value to high values allows more blurred images, however, number of total captured images increases. Decreasing the MMBM threshold results in less number of acquired images as seen in Table 6.2. Because, the time the metric spends below the preset threshold also decreases. Low metric thresholds may fail to capture any images when the robot moves so fast that real time MMBM never falls below the threshold. That was the case when the metric threshold was set



(a) (b)

Figure 6.5: Blob extraction from a blurry image. Blobs are seriously deformed and there exist false alarms. (a) The original image, (b) output of blob extraction.

(a) (b)

Figure 6.6: Blob extraction from a very blurry image. None of the blobs can be identified correctly and all of them are missed. (a) The original image, (b) output of blob extraction.

to 0.4 and 0.2 for the robot's fast walk mode and results were not included in Table 6.2. However, when the metric threshold is set properly, the number of images on which all blobs can be extracted increases compared to the uniformly sampled image acquisition. Furthermore, the threshold value can be made adaptive within certain range. For instance, it can be scaled with the minimum MMBM value in the last a few seconds. In other words, if the metric does not go below the threshold, it can be increased on the fly. Another adaptation approach could have been using the number of images captured in last couple of seconds. If the average frame rate drastically decreases, the threshold can be increased to capture slightly more blurred images. As a final remark, MMBM triggering significantly increases percentage of sharper images.

Table 6.2: Total Captured Images and Success Ratio of Blob Extraction

Walk Mode	**MMBM Th.**	**Walk Time**	**# of Fr.**	**# Missed**	**Success Rate**
Slow	N/A	35 sec	174	85	51%
Slow	0.8	35 sec	152	37	76%
Slow	0.4	35 sec	116	37	68%
Slow	0.2	35 sec	66	19	71%
Moderate	N/A	16 sec	78	39	50%
Moderate	0.8	16 sec	73	25	62%
Moderate	0.4	16 sec	56	10	82%
Moderate	0.2	16 sec	20	5	75%
Fast	N/A	5 sec	26	18	31%
Fast	0.8	5 sec	20	6	70%

Drop in the number of frames for MMBM based camera triggering is shown in Table 6.2. Capturing less number of images may seem like obtaining less information. However, average sharpness of images also increases which corresponds to information gain. Therefore, the number of captured images as well as their quality plays a crucial role on the total information gain. As clearly seen in Table 6.2, number of captured images gradually drops when MMBM threshold is set to lower values. Because, the time instances when MMBM drops below the threshold becomes more and more sparse and camera waits more for a suitable capture instance. In an ideal case, exposure time, t_e, dominates the time slot allocated for capturing one frame (t_e, t_r and t_f as shown in Fig. 5.1) in high fps video capture. Hence, any delay in the image capture means irreversible image loss and sampling rate drop. The experiment

designed in this section expresses that part of the problem too. However, there can be cases where a camera is required to work in low sampling rate. Thus, idle time of camera, t_f, would be relatively large even in fixed sampling rate capturing mode such as the one shown in this section. Large t_f gives the opportunity to advance triggering instances of camera, instead of delaying it. By delaying the trigger for suitable MMBM instances and advancing it (i.e. keeping t_f as small as possible) the trigger instances can be modified to lose less number of frames as well as having keeping the sharpness of MMBM based triggering. In the best case, number of captured images with MMBM based triggering can be the same as the number of images captured with uniform image acquisition.

CHAPTER 7

CONCLUSION

In this thesis, a new metric, MMBM, to evaluate camera egomotion based average motion blur of an image was presented. Detailed derivation of it is given for rotation only and free motion scenarios. MMBM can be computed for 6D motion of camera. But, it is evaluated by using only gyro data in real-time for the experiments. MMBM calculates optical flow vectors using only velocity of camera. In order to validate our derivations MMBM is compared to a conventional optical flow algorithm. Although the definition of MMBM involves an integral that can not be solved analytically, Riemann Sum approximation is introduced to calculate it in real time. Different application areas of MMBM are discussed. It is possible to use MMBM offline to have motion blur level of a captured image without performing any image processing. But, the main application pronounced in this work is externally triggering a camera to capture images at favorable time instances, i.e. MMBM based camera triggering, to capture sharper images and avoid extremely blurred ones. 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 and when it is monotonically decreasing. 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. Furthermore, the implications of obtaining sharper images on visual perception are explained with an experiment in which approximately 30-35% increase in feature extraction performance is shown.

An industrial grade camera was was externally triggered throughout the experiments.

That method was preferred to precisely control image capture instances and to start image acquisition as soon as required. However, it can be possible to use a regular camera as long as image acquisition can be started within a few milliseconds when a camera which can be externally triggered is not available. If a capture can be quickly started by sending commands to the camera through the data transfer cable, it can be conveniently used for MMBM applications.

MMBM can be used in many robotics applications that suffer from motion blur, legged robots in particular. Exploiting quasi-periodicity of a legged robot's body oscillations is the fundamental requirement for the applicability of MMBM based camera triggering. If the camera is moving at constant speed, all of the possible trigger instances would result in the same amount of motion blur. For example, MMBM based camera triggering can not improve image quality of an on board camera of plane, on the other hand, it can be conveniently used for helicopters or quadcopters when they are dynamically maneuvering on a workspace.

Further improvements can be achieved by incorporating the motion model of a robot or using the camera on a platform whose motion can be controlled and predetermined. Unfortunately, SensoRHex currently does not have any such motion models and its motion can not be rigidly controlled due to complex leg-ground interactions. If quasi periodicity of signals can be learned, predicting near future can be possible with signal prediction methods. Fitting a model to a quasi periodic body oscillations is still possible. But, the model would be pretty much limited. For instance such a signal model can be fitted only when the same input signal is applied to the robot and output oscillations does not considerably differ during experiments.

The goal to start this research is definitely achieved, even though there are still improvements that can be pushed further. Significant amount of motion blur can be successfully avoided for SensoRhex's working conditions by using our method , before it corrupts an image. Even though, only SensoRHex is demonstrated as an experimental platform, the mentioned methods can be applied to any camera working under oscillatory condition. For example, wearable cameras can be a prominent application area of MMBM since similar running behaviours would also affect camera performance of such devices. Yet another application can be implemented on mobile phones.

The remaining amount of motion blur can be deblurred using methods available in the literature after using MMBM basd camera triggering. For instance gyro data used to calculate MMBM can also be used to construct PSF for deconvolution based motion blur removal. Deblurring algorithms also suffer from extremely blurred images since the information loss would be on an irreversible level. Hence, MMBM can be used as a safety pre-process of deblurring algorithms.

Predictive MMBM implementation would have been a prosperous contribution in this work. Another demonstration which also includes translational motion of camera could have enriched the experiments.

REFERENCES

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

[2] 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*, pages 1547–1552, 2012.

[3] Hyeoungho Bae, Charless C Fowlkes, and Pai H Chou. Patch mosaic for fast motion deblurring. In *Computer Vision–ACCV 2012*, pages 322–335. Springer, 2013.

[4] A. Basu and K. Ravi. Active camera calibration using pan, tilt and roll. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(3):559–566, 1997.

[5] Peter Corke, Jorge Lobo, and Jorge Dias. An introduction to inertial and visual sensing. *The International Journal of Robotics Research*, 26(6):519–535, 2007.

[6] Shengyang Dai and Ying Wu. Motion from blur. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

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

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

[9] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. Manchester, UK, 1988.

[10] J. M. Hilkert. Inertially stabilized platform technology concepts and principles. *Control Systems, IEEE*, 28(1):26–46, 2008.

[11] Jing Hu, Yezhou Li, Shaozhang Niu, and Xianzhe Meng. Exposing digital image forgeries by detecting inconsistencies in principal point. In *Computer Science and Service System (CSSS), 2011 International Conference on*, pages 404–407, 2011.

[12] Junichi Ido, Yoshinao Shimizu, Yoshio Matsumoto, and Tsukasa Ogasawara. Indoor navigation for a humanoid robot using a view sequence. *The International Journal of Robotics Research*, 28(2):315–325, 2009.

[13] Neel Joshi, Sing Bing Kang, C. Lawrence Zitnick, and Richard Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29(4):30:1–30:9, July 2010.

[14] G. Klein and Tom Drummond. Robust visual tracking for non-instrumental augmented reality. In *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, pages 113–122, 2003.

[15] Heung Cho Ko, Mark P Stoykovich, Jizhou Song, Viktor Malyarchuk, Won Mook Choi, Chang-Jae Yu, Joseph B Geddes Iii, Jianliang Xiao, Shuodao Wang, Yonggang Huang, and John A. Rogers. A hemispherical electronic eye camera based on compressible silicon optoelectronics. *Nature*, 454(7205):748–753, 2008.

[16] G. Kricorissian. Motion induced blur minimization in a portable image reader, October 27 2005. US Patent App. 10/831,861.

[17] Y. Lee and Y.K. Yoon. Device for image stabilization of camera module, April 20 2006. US Patent App. 11/063,162.

[18] H. Lerman. Optical tracking and stabilizing system with a gimbal mounted mirror for establishing a line of sight, April 20 1993. US Patent 5,203,220.

[19] Anat Levin, Peter Sand, Taeg Sang Cho, Frédo Durand, and William T. Freeman. Motion-invariant photography. *ACM Trans. Graph.*, 27(3):71:1–71:9, August 2008.

[20] Huei-Yung 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*, volume 1, pages 461–467, 2005.

[21] Y. Liu. Hand-supportable digital-imaging based code symbol reading system supporting motion blur reduction using an accelerometer sensor, December 4 2012. US Patent 8,322,622.

[22] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[23] M. Mutlu, A. Saranli, and U. Saranli. A real-time inertial motion blur metric: Application to frame triggering based motion blur minimization. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, to appear.

[24] S.K. Nayar and M. Ben-Ezra. Motion-based motion deblurring. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):689–698, 2004.

[25] 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*, pages 2375–2380, 2010.

[26] E. Poon, G. Fu, and I. Clarke. Determining maximum exposure time to limit motion blur during image capture, March 24 2009. US Patent 7,509,038.

[27] Minu Poulose. Literature survey on image deblurring techniques. *International Journal of Computer Applications Technology and Research*, 2(3):286 – 288, 2013.

[28] 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*, pages 2250–2257, 2009.

[29] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, et al. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, pages 10823–10825, 2008.

[30] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[31] Uluc Saranli, Martin Buehler, and Daniel E. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

[32] Sebastian Schuon and Klaus Diepold. Comparison of motion de-blur algorithms and real world deployment. *Acta Astronautica*, 64(11-12):1050–1065, 2009.

[33] M.J. Smith, A. Boxerbaum, G.L. Peterson, and R.D. Quinn. Electronic image stabilization using optical flow with inertial fusion. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1146–1153, 2010.

[34] R. Sobol. Intelligent motion blur minimization, December 5 2002. US Patent App. 09/872,076.

[35] D. Stanvely, C. Whitman, G. Hofer, D. Campbell, and J. Yost. Image-stabilization systems and methods, November 10 2005. US Patent App. 10/842,223.

[36] D. Stavely. Apparatus and method for reducing image blur in a digital camera, August 16 2007. US Patent App. 11/786,210.

[37] D. Stavely, C. Whitman, G. Hofer, D. Campbell, and J. Yost. Image-exposure systems and methods, November 10 2005. US Patent App. 10/842,222.

[38] Deqing Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439, 2010.

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

[40] Y. Wakui. Digital camera with attitude and shake detection, August 2 2005. US Patent 6,924,837.

[41] T. Ziemkowski. Avoiding image artifacts caused by camera vibration, June 7 2007. US Patent App. 11/294,782.