Reducing Motion Artifacts in Mobile Vision Systems via Dynamic Filtering of Image Sequences

Christoph Walter, Felix Penzlin, and Norbert Elkmann

Abstract. Artifacts like motion blur are a common problem for vision systems on mobile robots, especially when operating under low light conditions. In this contribution we present a system that increases the average quality of camera images processed on resource-constrained mobile robots. We show a solution for estimating the magnitude of motion artifacts for every element of a continuous stream of images using data from an inertial measurement unit. Taking estimated image quality into account we describe an effective solution for congestion control between acquisition and processing modules. We build that upon a middleware that supports flexible flow control at a per-image level.

Keywords: robot vision, motion blur, congestion control, mobile robot.

1 Introduction

While the presence of motion artifacts in images from moving cameras can also be exploited in several ways, it is usually a troublesome effect. Objects may become unrecognizable because of blur; self-localization may yield poor results because of geometric distortion, and so on. At the same time, image processing tasks usually require significant resources and may easily exceed the capabilities of the computer hardware present on a mobile robot. In the following sections we describe our approach to lessen the effects of both problems. At first we discuss motion artifacts in more detail. After discussing related work we present our data processing scheme including approaches to image quality estimation and congestion control. We also present results achieved with our system.

Christoph Walter · Felix Penzlin · Norbert Elkmann
Fraunhofer Institute for Factory Operation and Automation, Sandtorstraße 22,
39106 Magdeburg, Germany,
e-mail: {christoph.walter,felix.penzlin,
norbert.elkmann}@iff.fraunhofer.de

2 Motion Artifacts

Cameras acquire images by exposing a light-sensitive element for a given period of time. Camera movement while the sensor is exposed may result in a number of image artifacts. The type and intensity of these artifacts depend on the exposure time as well as on several other parameters of the camera. In this section we discuss a number of common artifacts, in particular motion blur and geometric distortion. We use a pinhole camera model. Lens distortion is considered to have a negligible impact and is therefore not modeled here.

2.1 Motion Blur

Motion blur can be induced by moving either objects in the camera's field of vision or the camera itself. For simplicity we consider only a static scene and disregard any moving objects.

2.1.1 Translation of Camera

We distinguish between two kinds of camera movement. On the one hand there is translation in direction of the optical axis; on the other hand there is motion in the plane orthogonal to that axis. In the second case, the magnitude of blur b on the image sensor for an object is in inverse proportion to the distance to the camera plane d_{cam} (See Eqn. 1).

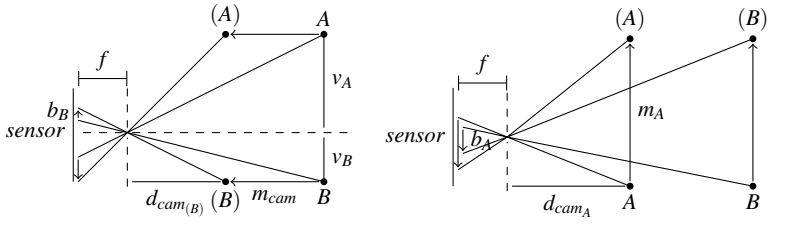

Fig. 1 Motion blur in case of vertical or horizontal translation (left) and translation parallel to the optical axis (right).

$$b = \frac{1}{d_{cam}} * m * f \tag{1}$$

For movements parallel to the optical axis the intensity of blur b for an object depends on its distances from line of view v and camera plane and the displacement d_{cam}. For a point at the optical axis, this kind of translation has no impact (See Fig. 1). If objects are relatively far away from the camera, translation becomes insignificant.

2.1.2 Camera Rotation

When rotating the camera, the magnitude of blur also depends on the position of a given object relative to the optical axis. Figure 2 shows that such a camera rotation results in a blur b roughly perpendicular to the axis of rotation. Its strength depends on the actual angle of rotation and on the angle between the rotation axis and the view direction. The distance to an object does not matter for rotational blur.

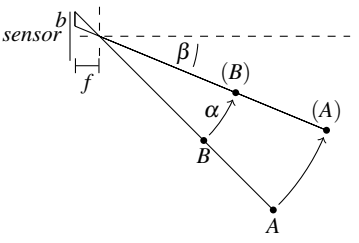

Fig. 2 Motion blur in case of rotation.

2.2 Distortion

Geometrical image distortion is another common artifact that can be found with moving cameras. It occurs when different portions of the image sensor are exposed sequentially to light. This mode is called "rolling shutter" and is implemented in various CMOS-based cameras.

Here, different phenomena can be observed. A sudden change in illumination may influence only portions of the image. If the camera is moved horizontally or rotated around the vertical axis, skew can be observed. Vertical lines appear to lean to the left for moving to the left or the right side for the opposite direction of movement. Vertical movements as well as rotations around the horizontal axis result in stretching respectively shrinking of objects vertically. Altering the direction of movement at a high speed (in case of vibrations) is called "wobble". When rotating the camera around the optical axis, straight lines get bent around the image center.

3 Related Work

Researchers have dealt with techniques to prevent, detect and remove motion artifacts in the past. There are also several commercial solutions available that implement such techniques. Instead of removing motion artifacts it would be beneficial to avoid them all together. The first approach means compensating the motion of the camera. Here, special hardware with accurate actuators is required. One solution is to stabilize the whole camera on a special platform, as shown in [11]. Similar platforms are available for cine camera mounted on helicopters or cars to countervail vibrations actively. Other solutions are shiftable image sensors ([13], [1], [2]) that counteract camera shake. Therefor an inertial sensor captures the movements of the camera to compute the appropriate shift. Instead of a shiftable sensor other

agile optical components can be used like a floating lens element, which is utilized in some digital SLR cameras, a variable fluid prism [10] or a movable mirror [7]. A simple solution for hand-held cameras tries to delay image exposure in case of camera shaking, which is determined using acceleration sensors.

A different class of solutions does not circumvent motion artifacts during acquisition, but tries to undo artifacts at a later point. For instance one approach ([4]) merges blurred and sharp but under-exposed images of the same scene to obtain an overall improved image. A number of algorithms for global shutter ([8], [6]) and for rolling shutter ([3], [9]) cameras have been developed. In general, a correction in software is time-consuming and consists of two steps. In the first step the artifacts are identified, in a second step these artifacts are removed from the image. Another method for dealing with motion artifacts is to use redundant information sources. Using this method is highly application dependent. For example: The authors of ([12]) have used data from inertial sensors in case of fast and camera-data in case of slow or no motion when tracking motion of humans.

4 Data Processing Scheme

While researchers made various efforts to deal with motion artifacts, many of them are not well suited for mobile robot applications. Evaluating the content of individual images is time-consuming. Undoing blur is even more computationally expensive and yields mediocre results in real life. Adaptive triggering of image acquisition depending on current camera movement is a promising and computationally inexpensive approach. However, it is a disadvantage that a sudden increase in camera movement during exposure cannot be predicted.

In our system, we chose to continuously acquire images as well as motion data and apply a selection process at a later stage. One advantage is that the actual movement of the camera during image exposure is known. Another advantage is that potential 'bad' pictures are not prevented but may be used in case of adverse camera movement for a prolonged period of time. We achieve such behavior by introducing an acceptance test for individual images that incorporates not only estimated image quality but also a system load indicator.

In this section we discuss our solution for the estimation of the image quality by predicting image artifacts through evaluation of angular rates of rotation of the robot. We also describe our approaches to system architecture as well as congestion control when using time-consuming image processing algorithms as data consumers.

4.1 Data Flow and Flow Control

We based the implementation of the image filtering approach on our existing CORBA-based [5] software architecture. Its main focus is on data flow and distributed processing. In our system, an application is constructed by combining software modules that provide functionality to other modules via a generic interface.

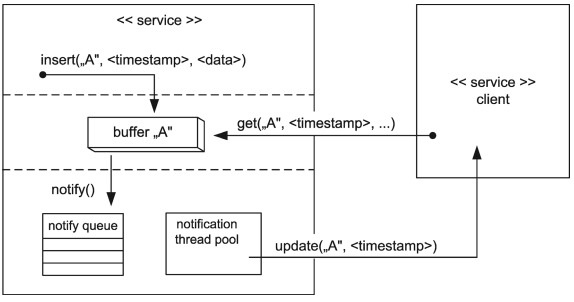

Fig. 3 A service provider offers data to clients via named buffers.

The modules are loosely coupled and may be arbitrarily distributed over a number of computers connected by a network. We call such a module a "service". Modules that only access functionality offered by other services are called "service-clients".

The interface of the services is data-centric. It allows services to offer data via named buffers to other services or clients. Each buffer represents a time series of specific data objects (See Fig. 3). A service can represent a data source, a processing module or a data consumer. Services that represent sensors or sensor systems may provide data that contains a single measurement, a time series of measurements, a vector, an image, or any other type of data that is generated by the sensor system. The data flow of the application is modeled by connecting the services via the publish/subscribe paradigm. In this communication model, the services play different roles. Please note that in our system a client does not subscribe to specific data elements but to notifications on when such elements become available. We allow the subscriber to specify a notification queue size. This allows the data source to produce a certain number of data elements in advance, even though the subscriber is still busy processing a previous element. Flow control is achieved by preventing a service from generating new data elements in case the notification queue gets full.

The data objects that are being exchanged between services satisfy a given scheme. They consist of binary data and metadata in XML syntax. The most important element of the metadata is the timestamp, which indicates the recording time of the original data. The timestamp acts as a key for the data object within the given data buffer. The binary data can contain information in any format. A client can request individual data elements or arbitrary blocks of data elements from a buffer (See Fig. 4). This is achieved by sending structured queries with a simple syntax

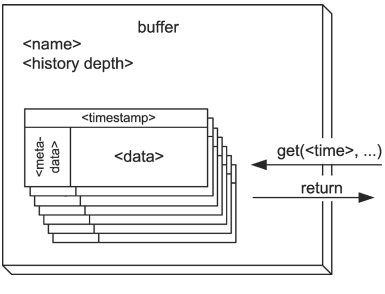

Fig. 4 Structure of a single buffer.

to the desired service. The client can then process these data items and/or eventually offer the results to further clients. The foundation of the used timestamps is the Newtonian time model. Timestamps are generated based on synchronized local clocks.

4.2 Image Quality Estimation

As discussed in 2, camera translation results only in marginal artifacts for objects at medium or large distances, which we consider dominant in many mobile robot scenarios.

For a camera mounted in the middle of the robot with a wheelbase of 0.5 m climbing an obstacle of 10 mm height the rotation results in motion artifacts for an object in a distance of 2 m seven times greater than by the translation component.

Hence we can simplify the tracking of camera movement by measuring rotation only. Angular rate gyroscopes were used for that matter. Because the strength of artifacts b depends on the angle between rotation axis and view direction, it varies across the image. In order to be able to decide, whether an image is too disturbed to be passed to image processing or not, we need a single quality estimate for the entire image. Several possibilities exist to calculate such an estimate, including the average or the maximum strength of motion artifacts of all regions of the image. We decided to go with another approach and focus on the region around the immediate image center only (See Eqn. 2).

$$b = \sum_{i=1}^{shutter} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{2}$$

Reasons that support that decision are: (1) cameras are usually arranged so that objects of interest are located near the image center; and (2) it leads to further simplification of the tracking of camera movement. Rotation (roll) around the optical axis of the camera does not contribute to artifacts at the image center. Hence we only need to measure rotation (yaw and pitch) around two axes parallel to the image plane. This setup gives a good estimation of the projection of the actual rotation axis onto the image plane if the distance between the camera's pinhole and the gyroscope unit is small. The strength of motion artifacts can then be calculated in a trivial way by integrating rotation present during image exposure.

To express the image quality $q \in (0, 1]$ we map the strength of motion artifacts b to the interval from 0 to 1, where 1 stands for an immobile camera. The quality reaches 0 for an infinitely high artifact strength (See Eqn. 3).

$$q_{image} = \frac{1}{e^b} \tag{3}$$

4.3 Image Acceptance Test and Congestion Control

We consider mobile platforms to have in general a limited computing capacity. At the same time we assume that image processing tasks consume significant resources. This is especially true if multiple image processing tasks are running simultaneously. Examples are visual SLAM, marker detection, terrain classification, object recognition, and so on. Here, the frame rate of a camera may easily exceed the image processing capacity. The system becomes congested and individual frames have to be dropped.

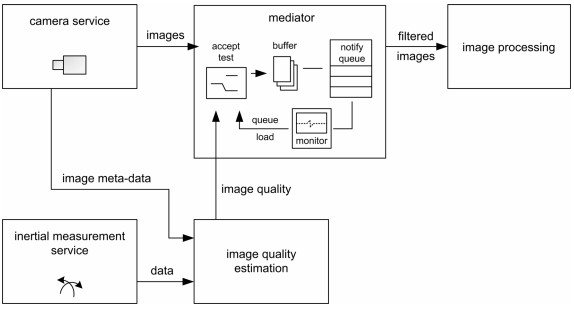

Fig. 5 Structure of a system containing a mediator service for congestion control.

Instead of dropping random frames, we can apply a simple, yet effective congestion control scheme. Having a normalized image quality value available for every frame enables us to compare it directly to the current system load, which is determined by the amount of pending images. Images are only accepted for further processing if their image quality is greater than the system load indicator. This implies that in case the system is idle, even images containing heavy motion artifacts are accepted for further processing. This ensures that it is not possible to reject every image for an extended period of time. However, a minimum required quality level can also be introduced if needed. The implementation shown in Figure 5 is self-explanatory. We chose to implement the acceptance test in form of a mediator that is arranged in between the data source (camera service) and a client that implements the image processing task. The system load can easily be computed by monitoring the current size of the notification queue as described earlier in this section. It is then normalized to the same interval as the quality.

5 System Evaluation

In this section we show results achieved with our approach to image quality estimation. We also present improvements of a scenario where markers are to be detected by a mobile robot while driving on a bumpy floor.

5.1 Evaluation of Motion Artifact Detection

In a first experiment we checked the correlation between motion blur in camera images and the calculated quality based on angular rate measurement. A mobile robot was equipped with a front facing camera and was driving towards a board placed in front of it. We did various measurements at different lighting conditions ranging from 300 lux to 600 lux, with a shutter time between 60 ms and 30 ms. The robot passed various bumps of a maximum height of 1 cm, which resulted in displacement as well as rotation of the robot. Two examples of images of different quality are shown in Figure 6. In image A (left) a strong motion blur in vertical direction can be observed, which resulted in a low quality estimate. Image B is significantly sharper, hence produces the better quality value. In Figure 7, the corresponding measurements from the gyroscopes are depicted. The intervals of image exposure are shown in gray.

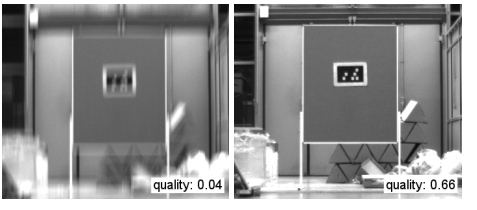

Fig. 6 Example images containing only slight (right) and strong motion blur (left).

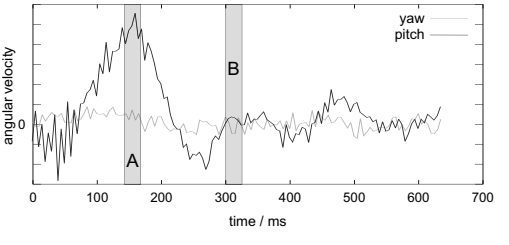

Fig. 7 Measured angular rates of rotation during image exposure (in arbitrary units).

For Figure 8 we compared the computed quality values 50 images to blur near the image center. The values for blur radius were generated by manual measurements.

5.2 Improving a Marker Detection Scenario

We applied our approach to dynamic image filtering to a scenario where optical markers were to be recognized by a moving mobile robot. The computing capacity onboard the robot is limited. Therefore not all images acquired by the onboard camera can be processed. The lights were set to about 300 lux, which resulted in an average integration time of 60 ms. A 2/3 inch monochrome CCD sensor and a

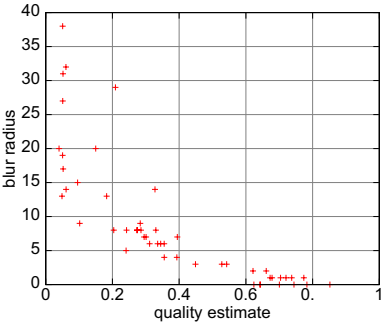

Fig. 8 Correlation between quality estimate and actual strength of motion blur.

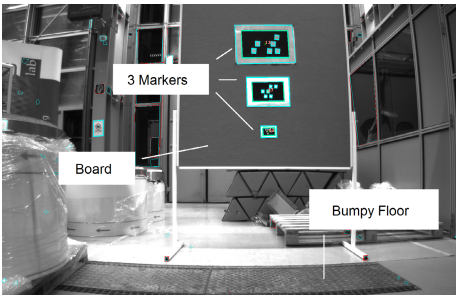

Fig. 9 Test setup of the marker detection scenario.

4.8 mm fixed focus C-Mount lens were used in the experiment. The iris was opened completely to allow as much light as possible to pass through. The robot was approaching a board from a distance of approximately 13 meters. Markers of different sizes were attached to the board (See Fig. 9). The goal when approaching the board was to recognize the markers as often as possible.

Images and motion data were recorded in order to compare results achieved with uncontrolled frame drops against results with dynamic filtering. In case of uncontrolled frame drops, the image quality was not taken into account. Here, the acceptance test was only considering queue utilization. The total number of images acquired during the approach was 319. The average processing time per frame required by the marker detection algorithm was approximately 1.6 times the inter-arrival time of new images. Table 1 shows the number of images in which a marker could be identified for one particular approach. The results were consistent when repeating the experiment. Please note that the largest marker was only visible in 246 out of 319 images. This is because it went out of view when the robot came close to the board. In general, markers could not be recognized in all 319 frames because at first they were too far away or they were obscured by motion blur. In the first run every image has been analyzed by the marker detection regardless of processing time. In the next run arriving images were dropped whenever the image-buffer was used to full capacity. In the third run images were discarded in the discussed manner.

It can be seen that the improvement in the total number of images with recognized markers increases with the decreasing size of the marker. This is because smaller markers are harder to recognize and are easily obscured by blur.

Table 1 Improvements of recognition results when applying dynamic filtering.

Marker	Images with marker recognizable	Marker recognized (uncontrolled)	Marker recognized (filtered)	Improvement (pct.)
large	246	154	161	4.5
medium	272	159	178	11.9
small	99	62	71	14.5

6 Conclusion

Here we presented an approach to improve the performance of image recognition tasks on mobile robots equipped with robust low-cost cameras. Concrete improvements have been achieved for an optical marker detection scenario. The basic idea presented was to improve the quality of processed images by estimating the amount of included motion artifacts for every image and rejecting bad ones. Our system is well suited for resource-constrained robots where the camera's frame rate exceeds the processing capabilities of the onboard computer. Based on our results, we are confident that the performance of a number of different image processing tasks can be improved through this approach.

Acknowledgements. This work has been funded in part by the German Federal Ministry of Education and Research under grant 01IM08002.

References

1. Cardani, B.: Optical image stabilization for digital cameras. Control Systems Magazine, IEEE 26(2), 21–22 (2006)
2. Chiu, C.W., Chao, P.P., Wu, D.Y.: Optimal design of magnetically actuated optical image stabilizer mechanism for cameras in mobile phones via genetic algorithm. IEEE Transactions on Magnetics 43(6), 2582–2584 (2007)
3. Cho, W.-H., Hong, K.-S.: A fast cis still image stabilization method without parallax and moving object problems. IEEE Transactions on Consumer Electronics 54(2), 197–205 (2008)
4. Choi, B.D., Jung, S.W., Ko, S.J.: Motion-blur-free camera system splitting exposure time. IEEE Transactions on Consumer Electronics 54(3), 981–986 (2008)
5. (eds.), O.M.G.: Common object request broker architecture (corba/iiop)
6. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. Graph 25, 787–794 (2006)
7. Gunthner, W., Wagner, P., Ulbrich, H.: An inertially stabilised vehicle camera system - hardware, algorithms, test drives. In: 32nd Annual Conference on IEEE Industrial Electronics, IECON 2006, pp. 3815–3820 (2006)
8. Ji, H., Liu, C.: Motion blur identification from image gradients. In: IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2008, pp. 1–8 (2008)
9. Nicklin, S.P., Fisher, R.D., Middleton, R.H.: Rolling shutter image compensation. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X. LNCS, vol. 4434, pp. 402–409. Springer, Heidelberg (2007)

10. Sato, K., Ishizuka, S., Nikami, A., Sato, M.: Control techniques for optical image stabilizing system. IEEE Transactions on Consumer Electronics 39(3), 461–466 (1993)
11. Schiehlen, J., Dickmanns, E.: Design and control of a camera platform for machine vision. In: Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 1994. Advanced Robotic Systems and the Real World, IROS 1994, vol. 3, pp. 2058–2063 (1994)
12. Tao, Y., Hu, H., Zhou, H.: Integration of vision and inertial sensors for 3d arm motion tracking in home-based rehabilitation. Int. J. Rob. Res. 26(6), 607–624 (2007)
13. Yeom, D., Park, N., Jung, S.: Digital controller of novel voice coil motor actuator for optical image stabilizer. In: International Conference on Control, Automation and Systems, 2007. ICCAS 2007, pp. 2201–2206 (2007)