An Outdoor Stereo Camera System for the Generation of Real-World Benchmark Datasets with Ground Truth

Stephan Meister, Bernd Jähne, Daniel Kondermann

Heidelberg Collaboratory for Image Processing
Interdisciplinary Center for Scientific Computing

Abstract. In this report we describe a high-performance stereo camera system to capture image sequences with high temporal and spatial resolution for the evaluation of various image processing tasks. The system was primarily designed for complex outdoor and traffic scenes which frequently occur in the automotive industry, but is also suited for other applications. For this task the system is equipped with a very accurate inertial measurement unit and global positioning system, which provides exact camera movement and position data. The system is already in active use and has produced several terabyte of challenging image sequences which are available for download.

1 Introduction

In many areas of computer vision and image processing well known test sequences are used to evaluate and compare algorithms and methods. Along with the availability of source code these are the most important elements to guarantee the reproducibility of scientific results. Many test sequences are usually tailored for a specific problem and are restricted to engineered scenes for example with static illumination. For many real world applications this is a highly unlikely setup. Take for example driver assistance systems which need to process images taken under dynamic lighting and weather conditions. For those applications few generic test sequences are available and if they exist they show only a small subset of all possible real-world effects which have to be dealt with. For the sake of designing robust image processing algorithms we need a method to produce challenging data, which covers the whole bandwidth real-life applications have to cope with.

To this end we have developed a stereo camera system which allows to produce sequences of complex outdoor scenes with a high spatial and temporal resolution. The system is tailored for mobile use inside a car, to record traffic situations which combine many challenging image processing aspects. By employing a high-precision GPS a recording can be started several times at the same coordinates but under different exterior conditions. This allows us to evaluate how the results of various algorithms can differ on the same scene.

1.1 Related Work

Single test sequences or whole image databases for a certain application are quite common. Examples for optical flow evaluation are the Yosemite sequences [1], the marbled block sequence [2], or the middlebury datasets [3]. Other examples are the PASCAL Visual Object Classes Challenge [4] or the middlebury stereo datasets [5]. Most of these are synthetic or indoor scenes with fixed illumination. More generic datasets which are not restricted to a single application are harder to come by. One example is the machine learning data set repository (mldata)[1]. Narasimhan et al. [6] have captured the same city scene over a timespan of several hundreds of days, providing comparison data for different weather and illumination conditions. In a similar manner Teller et al. created a database of calibrated images in a complex urban scene [7]. Liu and Klette [8] performed stereo and motion analysis on multiple traffic scenes out of a moving car. Geiger and Kitt et al. [9, 10] have produced similar stereo datasets with odometry data[2]. We aim to combine the stereo and automotive aspects with increased sequence sizes to produce superior datasets for evaluation purposes.

2 Stereo Camera System

First we will to summarize the capabilities of current hardware and some of the properties of commonly used evaluation sequences.

Most consumer cameras, but also many industrial or research systems operate at rates of 25 or 30 frames per second. This may be sufficient for computer graphics due to human perception, but many applications have higher requirements. In traffic scenes, where objects can move at several dozen meters per second, those cameras can produce severe motion blur (depending on exposure time) or temporal aliasing effects. Take for example wheels which appear to rotate backwards if their angular velocity is above a certain value. As a matter of fact optical flow estimation is mostly carried out on only two consecutive images [3]. Most sequences for evaluation purposes are therefore only a couple of frames long. On the other hand, algorithm design could benefit from new test sequences several hundred frames long. One example would be the integration of camera position data over several frames to increase its accuracy.

When computational cost is a factor, the spatial resolution is often the first victim, especially when real-time performances in the range of 25 frames per second are required. Nevertheless is a high resolution beneficial for many algorithms, be it for the depth resolution in stereo vision or the distinctiveness of features. With the field of computer vision in mind, it is desirable to work with resolutions close or similar to current display technologies (e.g. HDTV).

Furthermore, many algorithm evaluations are carried out on 8bit grayscale or color images. This is but merely a historic remnant as earlier cameras and image formats were restricted to this bit depth and display still are. (Although high

[1] www.mldata.org

[2] www.rainsoft.de/software/datasets.html

depth displays are available for medical and print applications) But nowadays, even cheap consumer cameras have internal bit depths of 10, 12 or even more. Therefore there are few excuses not to use cameras with a high bit-depth and to profit from the increased amount of information.

With this in mind we composed the capture system out of the following components: [3]

Hardware

We use two Photonfocus MV1-D1312-160-CL CMOS cameras, which deliver 100 frames per second at an resolution of 1312x1082 pixels ($8 \mu m x 8 \mu m$ per pixel) and 12 bit depth, to acquire the images. Additionally, they are equipped with global shutter to reduce artifacts due to fast motion and are practically bloom-less, which is an advantage in fast changing lighting conditions.

For the lenses we use Linos MeVis-C high precision lenses with a focal length of 25mm and a maximum radial distortion of less than 0.3 %.

Apart from the raw image capturing components, a high precision NAV440 Navigation System by Crossbow [4] is also part of the system. The NAV440 combines a global positioning system (GPS) receiver with an inertial measurement unit (IMU).

Heading and acceleration data are provided with an accuracy of $< 1.0^\circ$ and $< 9.8 \cdot 10^{-3} \frac{m}{s^2}$ while the system operates at frequencies of up to 100Hz reaching the same speed as the cameras.

The GPS data enables us to start the image capture process at precisely determined points, so that real-world image sequences taken at the same position but under different lighting (or e.g. weather) conditions can be compared.

The cameras are connected to a standard desktop PC, which is optimized for low power consumption of $\lesssim$ 120 W, to make it suitable for mobile use. Except for the storage system the PC has no particularly high processing power, although as the software makes heavy use of processing threads to satisfy various latency requirements, a processor with a least two cores is necessary.

At full speed the cameras reach a raw data rate of 541 MB/s. At the time of first construction the maximum data rates for harddisks reached values of 80-130 MB/s, depending strongly on interface, individual model and the actual region of the disk where write-operations occur. This last point is especially problematic as it represents a constraint on the minimum transfer-rate, not its average.

Write operations to a hard-disk are usually slower if sectors on the inner regions of the drive plater are accessed. Data fragmentation and space allocation policies of the file systems in use make this matter even worse. Modern Solid-State-Disks (SSD) seem promising, but long-term evaluations regarding their reliability and speed are not yet available.

[3] Special thanks to Martin Schmidt for selecting and assembling the hardware components and for initial works on the capture drivers and compression scheme.

[4] Crossbow Technology Inc., http://www.xbow.com/defense-solutions/products/NAV440.html

So, as no single storage medium is capable of the data rates needed, the system uses four of the fastest available hard disk drives (Western Digital Velociraptor WD3000GLFS) in an RAID 0 (redundant array of independent disks) configuration to spread write operations equally over all drives. This allows for maximum writing speed, but makes the system susceptible to data loss in case one of the disks is damaged.

Software

In any case a simple form of (lossless) data compression is necessary to make continuous writing of image data possible. One can exploit the fact that the four most significant bit of each 16bit word are zero, as the cameras provide only a pixel depth of 12 bit. By filling the gaps with values from other pixels the size of the data can be reduced to three fourth of the original amount.

Furthermore all file operations have to circumvent operating system write buffers to achieve maximum throughput. This is necessary because writing a file with standard library functions may induce unpredictable latency, depending on the number of bytes, the position of the memory buffer etc. So, instead the program uses the operating system low-level API to write directly to the filesystem. This imposes the additional constraint that write buffers need to be page-, as well as sector-aligned, resulting in the fact that write operations must always target multiples of a certain hardware-dependent number of bytes.

Another option would have been to ignore the filesystem completely and directly access the hard disks as block devices. However the performance impact was negligible in contrast to the increased burden of managing raw block data.

The acquisition application is written in C++ using the Qt Development Framework [5] in order to provide a graphical user interface, while the image processing part of the application depends heavily on the CImg library [6]. The operating system Microsoft Windows 7 (x64) is used. The current list of additional features which are usually not found in regular capture software reads as follows[7]:

- single Frame mode allows acquisition of single images or accumulated images for an arbitrary number of frames
- automatic fixed-pattern-noise reduction in single frame mode
- capture trigger over TCP/IP
- automatic start of sequence acquisition at predefined GPS-coordinates[8]
- automatic pausing of acquisition if vehicle speed drops below a certain threshold[8]

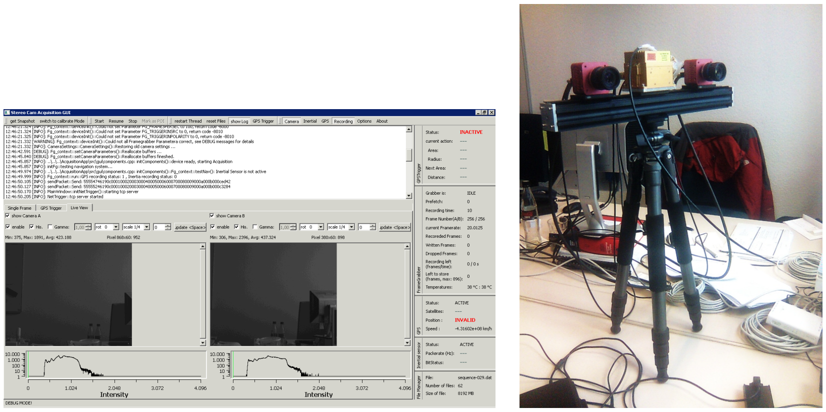
[5] Nokia, Qt Development Frameworks, http://qt.nokia.com/

[6] David Tschumperlé : The CImg Library: C++ Template Image Processing Toolkit, http://http://cimg.sourceforge.net/

[7] Additional thanks to Christoph Koke and Julian Coordts for the NAV440 implementation,GPS trigger code, extensive code refactoring and bugfixes

[8] Implemented by Christoph Koke

- custom file format for capturing simultaneously stereo images, camera parameters, GPS position, inertia data and additional meta data[8]
- circular memory buffer allows capturing of up to 12 seconds *before* the acquisition command was issued.
- tight integration with stereo rectification software.



```
12:46:21.324 [INFO]: Fg_context::deviceInit()::Could not set Parameter FG_FRAMESPERSEC to 100, return code -6000
12:46:21.324 [INFO]: Fg_context::deviceInit()::Could not set Parameter FG_TRIGGERINSRC to 0, return code -8010
12:46:21.325 [INFO]: Fg_context::deviceInit()::Could not set Parameter FG_TRIGGERINPOLARITY to 0, return code -8010
12:46:21.332 [WARNING]: Fg_context::deviceInit()::Could not all Framegrabber Parametera correct, see DEBUG messages for details
12:46:21.332 [INFO]: CameraSettings::CameraSettings()::Restoring old camera settings ...
12:46:42.591 [DEBUG]: Fg_context::setCameraParameters()::Reallocate buffers ...
12:46:45.840 [DEBUG]: Fg_context::setCameraParameters()::Reallocate buffers fineshed.
12:46:45.857 [INFO]: ..\..\..\AcquisitionApp\src\gui\components.cpp: initComponents()::device ready, starting Acquisition
12:46:45.857 [INFO]: initFg::testing navigation system...
12:46:49.974 [INFO]: ..\..\..\AcquisitionApp\src\gui\components.cpp: initComponents()::Fg_context::testNav(): Inertial Sensor is not active
12:46:49.999 [INFO]: Fg_context::run::GPS recording status: 1 , Inertia recording status: 0
12:46:50.105 [INFO]: sendPacket::Send: 55554746190c0001000200030004000500060007000800090000a000b000ced42
12:46:50.127 [INFO]: sendPacket::Send: 55555246190c0001000200030004000500060007000800090000a000b000c3284
12:46:50.170 [INFO]: MainWindow::initNetTrigger()::starting tcp server
12:46:50.205 [INFO]: NetTrigger::tcp server started
```

☑ show Camera A
☑ enable ☑ His. □ Gamma: 1,00 rot 0 scale 1/4 0
Min: 375, Max: 1891, Avg: 423.188 Pixel 868x60: 952

☑ show Camera B
☑ enable ☑ His. □ Gamma: 1,00 rot 0 scale 1/4 0
Min: 306, Max: 2396, Avg: 437.324 Pixel 380x60: 898

Status: INACTIVE
current action: ---
Area: ---
Radius: ---
Next Area: ---
Distance: ---

Grabber is: IDLE
Prefetch: 0
Recording time: 10
Frame Number(A/B): 256 / 256
current Framerate: 20.0125
Recoreded Frames: 0
Written Frames: 0
Dropped Frames: 0
Recording left (frames/time): 0 / 0 s
Left to store (frames, max: 896): 0
Temperatures: 38 °C : 38 °C

Status: ACTIVE
Satellites: ---
Position : INVALID
Speed : -4.31602e+08 km/h

Status: ACTIVE
Packerate (Hz): ---
BitStatus: ---

File: sequence-029.dat
Number of files: 62
Size of file: 8192 MB

DEBUG MODE!

(a) Screenshot of image acquisition software

(b) Stereo camera rig

Fig. 1: Camera System

3 Radiometric Camera Calibration and Spatial Sensor Non-uniformities

Handling of sensor noise is an important aspect of any non-purely synthetic image processing task.

Image sensors (both CCD and CMOS sensors) have different noise sources. Most obvious in cameras is the ***dark response non uniformity (DRNU)***, which is caused by (mostly thermal) creation of electron/hole pairs in the semiconductors in use. Its intensity depends on the temperature and exposure time.

By taking additional images with a closed aperture (or closed lens cap) one can sample over this noise and subtract it's mean from all captured images. Of course this does not take into account the temporal variation of this noise source, but this is usually below 0.5% of the maximum intensity once the chip is in a thermal steady state.

In a second postprocessing step the spatial ***photo response non-uniformity (PRNU)***, and the individual non-linear response curves of each pixel are dealt with. Different sensor elements will report different digital values for the same amount of light (photon current) falling onto it. CMOS sensors are more prone to this effect as each pixel has it's own amplifier with it's own amplification curve. For CCD sensors there can be quantitative differences between alternating rows. Also, the resulting gray-values deviate from an optimal linear response to the incident light. Figure 2 shows the response curve of two different pixels. Superimposed is a second order polynomial curve fitted to the data points. According to the EMVA 1288[9] standard for cameras and machine vision sensors this curve is temperature independent (only the constant offset changes with temperature) [11].

By mounting the cameras into an integrating sphere all of these photon-incidence to gray-value output curves were measured for 3 different wavelengths at every sensor pixel. For each incident photon current, the (temporal) arithmetic mean over 400 images was computed to get the response for each pixel. This was done for 200 different light intensities ranging from full darkness to an intensity where practically all pixels were saturated. For most of the pixels and intensities lower than $\approx 90\%$ of the saturation value, a second order polynomial fit describes the data quite well. For higher intensities the deviations usually increase, which is acceptable as the subtraction of the fixed pattern noise shifts the intensities to lower values. The Histogram of the root-mean-square-errors of the fits (Figure 3) shows that most pixels conform to our model but the number of pixels with higher deviations drops only slowly.

The polynomial function allows for calculation of the photon intensity for a given pixel gray-value. To compensate for defective pixels, a local 3x3 median filter was used to replace the values of those irregular pixels.

Defect pixels were detected by two methods: First all pixels whose gray-value is below a certain threshold in the integrating sphere image with maximum light intensity are marked. This involves complete dark or weak pixels, but not hot ones.

Second, all pixels which always differ significantly from their neighbors are marked. This is done by taking about 700 sample images from different scenes and checking for each pixel how its gray-value differs from the mean of its 4-neighborhood. Those pixels where the difference is above a certain threshold (100 in this case) get marked. If the pixel was marked at least 30 times in those 700 images it is finally marked as a dead/hot pixel.

Keep in mind that those pixels are not necessarily defective but may only have response curves which are slightly off the mean or are not described well by our sensor model. Dust on the sensor, which can not be completely avoided, is but one possible causes for this behavior. The results of the radiometric rectification can be seen in Figure 4, which shows a part of the sky.

[9] http://www.emva.org

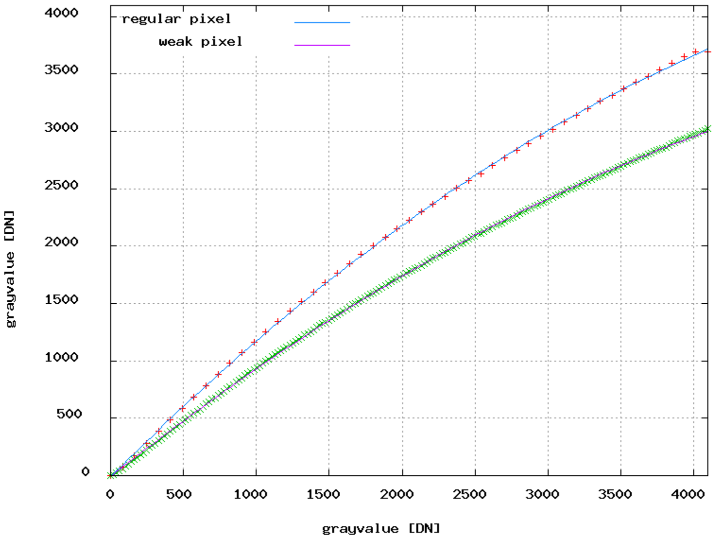


Fig. 2: photo response curve for a regular and a defective pixel (PRNU already subtracted)

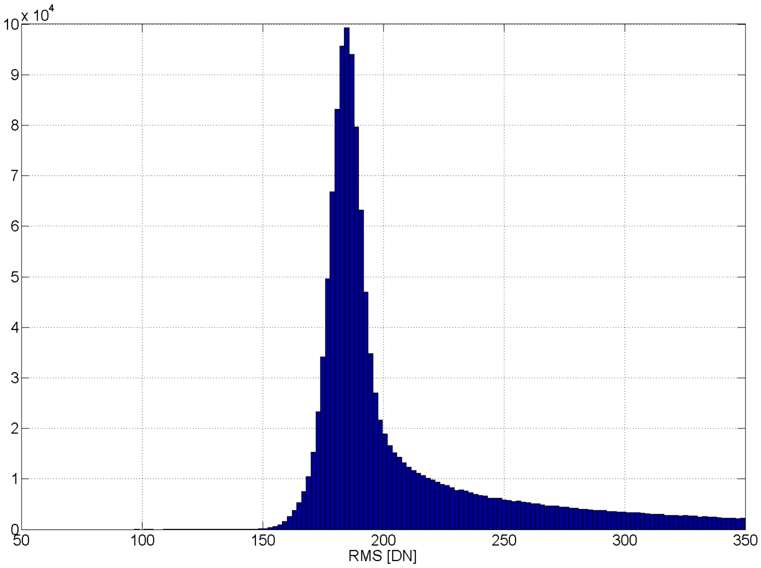
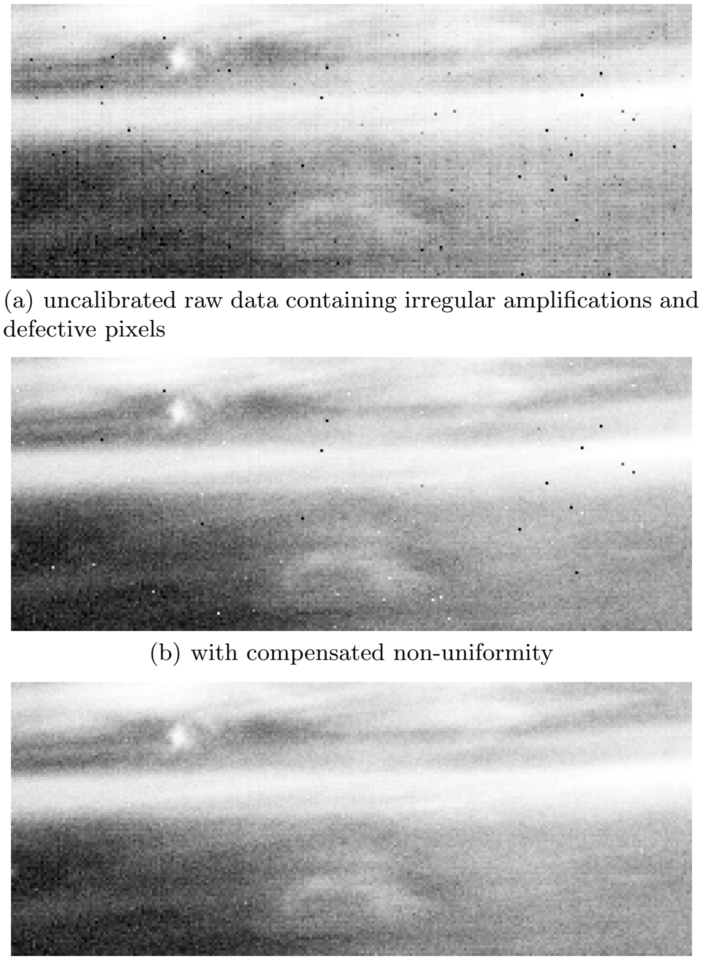


Fig. 3: Histogram of RMS errors of curve fits



(a) uncalibrated raw data containing irregular amplifications and defective pixels

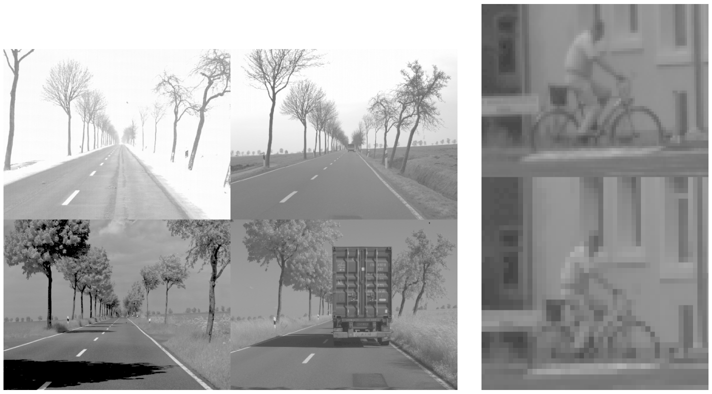
(b) with compensated non-uniformity

(c) with defective pixel compensation

Fig. 4: Results of radiometric image rectification (non-linear brightness scaling)

4 Usage and Example Applications

The camera system is already in active use. On 12 different dates we captured image sequences of about 15 seconds length at about 75 points in and around the German city of Hildesheim. This corresponds to about 10 TB of image data showing day-to-day driving situations like highway, inner-city, pedestrians or busy crossings under different lighting and weather conditions. Each of these short sequences is tagged with various keywords to allow a quick search for conditions which are of interest for a given application or algorithm. Example tags are *horizon, traffic sign, clouds, tunnel* etc. [10] Some Examples can be seen in Figures 5 and 6.



(a) same scene at different times

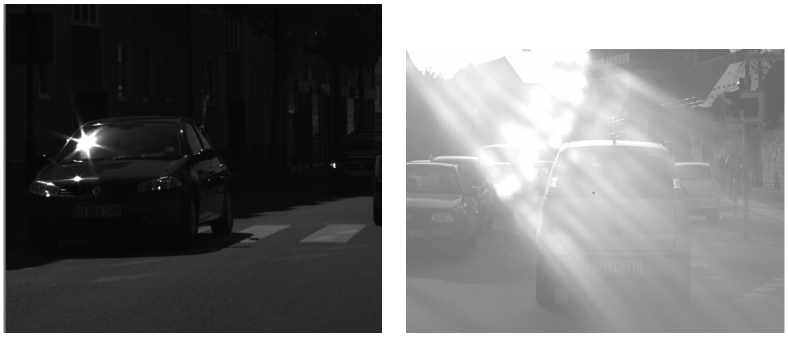
(b) improved resolution

Fig. 5: Example Situations

One exemplary usage of the data is described in [12]. By combining the stereo data with monoscopic camera tracking the optical flow for rigid scenes can be computed. An example flow field can be seen in figure 7(a). The method does not provide the high sub-pixel accuracy of state-of-the-art optical flow methods although many industry partners deem an one-pixel accuracy as completely satisfying for most applications. What they consider more important are robustness, illumination independence and predictable and improved corner behavior.

To provide this accuracy, certain upper thresholds for the disparity estimates and camera movement need to be uphold. These include 1-2 pixels accuracy for the disparity maps and at least 5 percent accurate camera translation estimation

[10] Parts of the sequences are available for download under http://hci.iwr.uni-heidelberg.de/Benchmarks/

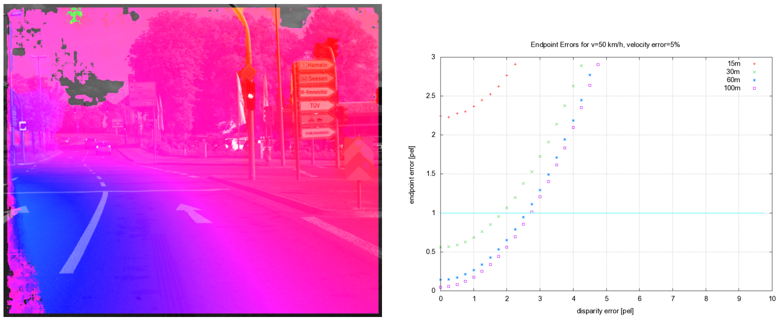


(a) intense reflections (b) reflections on windshield

Fig. 6: Example Situations

(with respect to the vehicle's speed). Expected flow endpoint errors for this configuration can be seen in figure 7(b).

We have not yet exhausted all the possibilities to improve the methods accuracy but the current results are already sufficient for many considered applications.

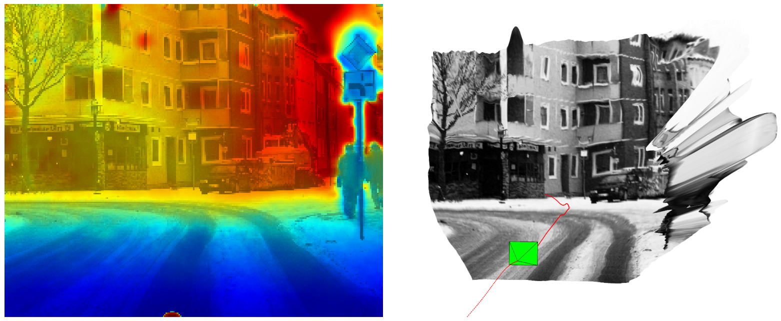


(a) Traffic scene with flow field as HSV overlay (b) expected endpoint errors for different distances from the camera

Fig. 7: Optical Flow calculation from depth and camera tracking

Another application is the robust reconstruction of 3D scenes recorded with a moving monocular camera. The method estimates the scene depth and external camera parameters (position and orientation), see Figure 8 for an illustration.

Such data can for example be used for obstacle detection, automatic distance keeping etc. (in autonomous mobile robots or in driver assistance systems).



(a) estimated depth as color overlay

(b) 3D reconstruction

Fig. 8: Robust 3D scene reconstruction

5 Conclusions and Further Work

We presented a high-performance stereo camera system specialized for the capture of high resolution outdoor sequences. It can produce images of about 1.4 Megapixels at 100 Hz supported by IMU and GPS data. Several terabyte of fully rectified (stereo and radiometric) image data for use in a evaluation database have already been produced and multiple projects which could use this data are planed or under active development. The modular design of our components and software allows us to improve the camera system easily in the future. Combining the system with color or Time-of-flight cameras to increase the robustness of depth estimates is only one of several future projects. Advances in general computer performance and small scale framegrabber devices could allow us to base the system on a laptop platform. This would allow us to fit the whole system into a backpack and produce evaluation sequences of nearly any region a human can access. Example applications for those sequences could be urban or indoor 3D reconstructions as well as research regarding augmented reality.

By pushing the complexity, versatility and size of existing evaluation sequences we hope to positively influence the further development of many image processing task.

References

1. Heeger, D.: Model for the extraction of image flow. Journal of the Optical Society of America **4**(8) (1987) 1455–1471
2. McCane, B., Novins, K., Crannitch, D., Galvin, B.: On benchmarking optical flow. Computer Vision and Image Understanding **84**(1) (2001) 126–143
3. Baker, S., Roth, S., Scharstein, D., Black, M., Lewis, J., Szeliski, R.: A database and evaluation methodology for optical flow. In: Proceedings of the International Conference on Computer Vision. (2007) 1–8
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html

5. Scharstein, D., Szeliski, R.: Middlebury stereo website. http://vision.middlebury.edu/stereo/
6. Narasimhan, S., Wang, C., Nayar, S.: All the images of an outdoor scene. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: Computer Vision ECCV 2002. Volume 2352 of Lecture Notes in Computer Science. Springer Berlin, Heidelberg (2002) 3–13
7. Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M., Master, N.: Calibrated, registered images of an extended urban area. International journal of computer vision **53**(1) (2003) 93–107
8. Liu, Z., Klette, R.: Performance evaluation of stereo and motion analysis on rectified image sequences. Technical report, Computer Science Department, The University of Auckland, New Zealand (2007)
9. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Asian Conference on Computer Vision, Queenstown, New Zealand (November 2010)
10. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In: IEEE Intelligent Vehicles Symposium, San Diego, USA (June 2010)
11. Erz, M.: Charakterisierung von Laufzeitkamerasystemen fr Lumineszenzlebensdauermessungen. PhD thesis, University of Heidelberg (2011)
12. Meister, S.: A study on ground truth generation for optical flow. Master's thesis, Heidelberg Collaboratory for Image Processing, University of Heidelberg (2010)