Ondrej PRIBULA, Jan ŠEDIVÝ, Jan FISCHER

Department of Measurement, Faculty of Electrical Engineering, Czech Technical University in Prague

Cost-effective image acquisition system for precise PC-based measurement

***Abstract**. Image sensors based on the CMOS technology became commonly used sources of image information in PC-based vision systems. Image information is converted directly on the chip to a stream of synchronous digital data which can be transferred to a PC using the cost-effective interface introduced in this contribution. The CMOS image sensor together with the developed interface can be considered as a simple image acquisition system – USB camera matching all features of distributed image acquisition and processing model described in this article. Concept of the USB camera is enhanced by additional data merged with each image frame transferred to the PC. This feature makes the developed device suitable for tasks where the precise timing of image acquisition together with an external synchronization is required. Based on the VHDL model of the used CMOS image sensor and the designed PC interface with the USB driver, the critical timing and maximal USB data throughput limitations are analyzed and discussed.*

***Streszczenie.** Informacja o obrazie z czujnika obrazu typu CMOS jest bezpośrednio przetwarzana w strumień transferowany do komputera. Kamera USB może być traktowana jako system pobierania i przetwarzania informacji. W artkule analizowany jest taki system, a szczególnie timing i szybkość przepływu danych. (**System akwizycji obrazu w pomiarach z wykorzystaniem komputera**)*

Keywords: Digital image processing; High-Speed USB interface; CMOS image sensor; Programmable logic device
Słowa kluczowe: cyfrowe przetwarzanie obrazu, czujnik obrazu.

Introduction

The fundamental part of all computer vision chains is the image acquisition unit. It is limiting many of the resulting system key features - especially the time of reaction and the error of measured parameters if the system is used for precise image-based measurement (e.g. physical dimensions of objects). Two basic factors are limiting the response time of measurement system: the maximal number of images (frames) which is the image source capable to acquire within the specified time and execution time of the used image processing algorithm. Number of recognizable image details, limiting the measurement precision, is given by a physical resolution of the used image source (image sensor).

Apart from the precise measurement tasks of computer vision, a camera interfaced to PC is becoming a fundamental part of many biological and bio-physical experiments. In such applications the image captured by the camera has to be synchronized with other measured signals to provide a valid optical reference. Based on these complex information behaviour of the observed object is evaluated (e.g. relationship between the neuro-electrical brain activity and physical movements of a laboratory rat).

In PC based image processing systems an input image can be acquired using two basic methods:

- Analog video camera is connected to a frame grabber unit where the analog video signal is converted to digital data and transferred to PC
- Image is acquired directly with a digital camera connected to PC

The first method, analog video signal digitalization, is losing its popularity mainly because of low resolution (maximum 768 horizontal and 576 vertical scan-lines for analog video in PAL standard) and worse overall image parameters given by the analog nature of transmitted signal (e.g. pixel jitter which can affect results of dimension measurement and measurement repeatability [1]). The frame grabber unit price may easily exceed the cost of the utilized analog camera. For simple applications with low requirements on image quality a TV tuner PC card can be used instead of the frame grabber [2].

The second way of image acquisition, most common in modern vision systems, is the digital image source (image sensor) connected to PC by a high-throughput interface, e.g. USB, FireWire or Ethernet (GigE). The CMOS (Complementary Metal Oxide Semiconductor) technology image sensors [3] are becoming market leaders in last years. Technological features, especially on-chip analog signal preprocessing and digitalization are determining them for cost-sensitive applications and consumer electronics design (e.g. web-cameras).

Compared with CCD (Charge Coupled Device) sensors CMOS image sensors have comparable physical parameters and in many applications can replace them. CMOS image sensor has in comparison to CCD sensor some functional and application advantages. The most significant feature is that image information is directly digitalized on the CMOS sensor chip with no need of additional electronic parts (analog-front-end for CCD [4]). It doesn't need sequential image information readout and allows theoretically random access to all pixel data [5]. Available CMOS sensors are equipped with so-called window readout mode. In this mode only the subset of pixels (selected window) is read out from the sensor what is increasing the frame rate. Image resolution can also be reduced by sub-sampling. In this mode the physical resolution of the image sensor is artificially decreased and only every N-th pixel is read out. Functions for automatic exposure and gain control can also be a part of electronics integrated on the CMOS image sensor chip.

In this article we introduce a newly developed USB interface for direct connection of CMOS image sensor to PC – a simple USB digital image acquisition system. The developed hardware solution consists of three parts: CMOS image sensor, simple CPLD-based (Complex Programmable Logic Device) glue logic and USB interface chip. Distributed image acquisition model, as a possible standard for direct fast image source to PC connection and low level data preprocessing, is introduced. Connecting of the CMOS image sensor as a fast synchronous data source with the PC has critical timing requirements. PC-latency-caused errors in the high speed image streaming are analyzed using a VHDL simulation model and results are presented in the following paragraphs.

Work background

The most reasonable image source for machine vision in terms of system cost can be a simple web-camera (webcam). This solution is satisfactory in many applications where the image format, compression, resolution, data rate and image synchronization are not crucial [6]. On the other hand, requirements on image quality for PC based vision systems intended for precise measurement task in industrial automation, testing or in R&D (Research and Development) cannot be easily satisfied by using standard webcams.

The low image quality of webcams is caused primarily by the use of lossy image compression which decreases the amount of data transferred from camera to the PC. For human eye the errors caused by compression and decompression are negligible and subjective image quality degradation is acceptable. In terms of image details recognition or precise measurement the resulting image after decompression is corrupted by artificial entities newly added to image (so-called compression artifacts) and by missing details which were removed by compression. Most common artifacts are block-like structures caused by compression, called blocking artifacts or blockiness [7]. A special case of blocking artifacts is blockiness on objects borders. It causes a virtual deformation of the measured object edges and results into an error in edges positions detection and increases thus the overall dimension measurement uncertainty [8]. Concerning the described compression features, uncompressed image data (raw data) are preferable for image processing algorithms focused on precise object shape and position evaluation.

For tasks where the acquired image information is a part of complex experiment results involving also other additional data (e.g. time-varying environment parameters or analog signals values) synchronization between the image and other data sources is crucial. Even if the data acquisition unit and camera are started synchronously, the delay caused by the different data paths in PC cannot be neglected (generally the delay is given by the chain beginning with the device driver and ending with task-scheduler of the operating system). In case of standard webcam or digital camera, which is sending pure image stream to PC, synchronization can be provided by an external element which change-of-state can be easily detected in both – the image and the additional data. Frequently used method is a LED (Light Emitting Diode) placed in the camera field of view. Flashing of this diode is controlled by external synchronization signal recorded also in the additional data (e.g. one bit in digital data). Image processing algorithm than detects the LED state and data are accordingly time-shifted and synchronized. This principle is used for example in bio-physiological research of laboratory rats behavior after direct brain electrical stimulation [9]. This method brings a time-synchronization uncertainty of range between zero and the frame exposition time (synchronization event can occur at the very beginning, during or at the end of frame exposition) what can be in some cases unacceptable. The described method is also unusable in tasks where the synchronization object cannot be a part of the observed scene, (e.g. when the object is observed through a microscope). Applicable and useful solution of synchronization is to provide image data with an additional information (e.g. digital synchronization which state is reported) sampled and transferred directly before or after the frame data (in a special case after each row of image data). An advantage can also be the option to synchronize the start of image streaming by an external signal, asynchronous frame time-stamping, snapshot mode or external lighter synchronization. The last two mentioned modes can take advantage if the camera is used together with a microscope [10]. The price of present digital cameras which provides some of image information synchronization methods described above is significant and their application for a cost-sensitive solution is impossible.

Inspired by the above mentioned disadvantages of existing digital cameras the new cost-effective solution of USB image acquisition device is described in the following chapters. The developed USB camera has modular structure which allows the use of almost any of existing CMOS image sensors with standard interface and to take advantage of all the sensor operation modes. PC interfacing is based on the newly developed distributed image acquisition model which allows high throughput image data transfers in the real time. Hardware solution of the designed camera enables advanced modes of image acquisition synchronization. Every transferred frame (or frame raw) contains besides the image data also general purpose arbitrary data section as the best option for data-to-image synchronization.

Hardware solution and VHDL model

While there are sensors providing the image in resolution of several megapixels and frame rates of several thousands of frames per second [11], the common sensor resolution vary from 300 kilopixels to 1.3 megapixels, with frame rates from 30 fps for larger sensors up to 60 fps for smaller sensors. When only 8 bits from the converted pixel value is transmitted, the required data bandwidth is 20 MB/s up to 40 MB/s. From the standard PC peripherals, only the high-speed USB 2.0 and FireWire are capable of delivering such bandwidth. While FireWire was designed for high-speed data transmissions, it is still not as common as USB and when the system cost is taken into an account, the USB interface comes out as the best solution. Furthermore, most computers are equipped with several USB ports and allow thus the connection of several independent cameras.

There are also several hardware implementation of USB 2.0 interface chip varying in given design freedom. For this system, the Cypress FX2 (CY7C68013A) controller was chosen as it provides internal buffers for load balancing and a small x51 microprocessor for configuration tasks. Even if it may be possible to connect the sensor directly to the Cypress chip or use a simple glue logic, such solution would bring no additional features to the system and the result will be just another web-cam providing an uncompressed image data. Furthermore the interfacing signals of the CMOS sensors differ slightly between sensor manufacturers and even, between different sensor types from the same manufacturer. Therefore the configurable platform with CPLD was employed in the final design depicted in figure 1.

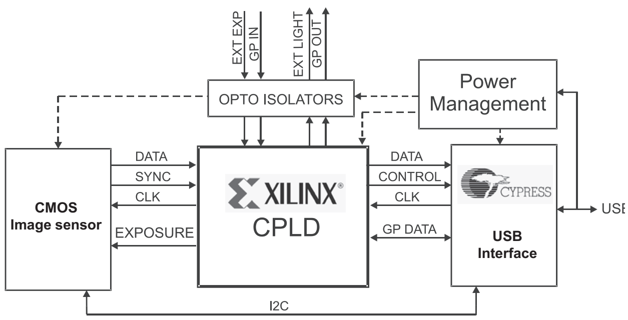

Fig.1. Block diagram of the designed image acquisition system

Besides the support of different image sensors, the programmable logic provides a perfect platform for image preprocessing and embedding of additional information. In such configuration, the image acquisition can be controlled by an external signal and correct timing can be assured by embedding an unique timestamp into each individual image. In horizontal blanking intervals after each line, the value of external signals can be transmitted with sampling frequency of 20 kHz and higher, and in frame header and footer, the simple image characteristics like minimal, maximal and average intensity can be transmitted, simplifying the later image processing performed on the PC.

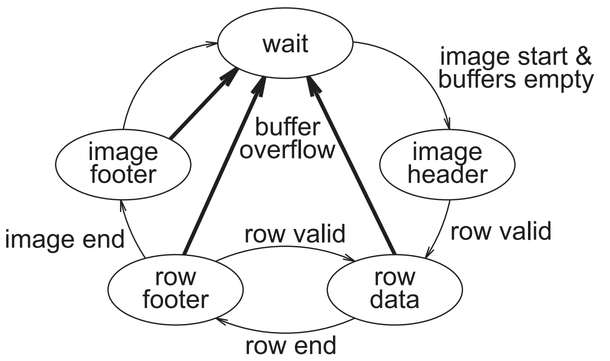

Fig.2. The basic CPLD core state diagram

The basic function of the core sensor interfacing logic is depicted in figure 2. The main purpose of this system is to ensure at least one image successfully transmitted over the USB even in case of no special driver implemented on the PC. At the beginning, the system waits for a new frame. When an active edge is detected on the sensor frame synchronization flag and the USB interface chip is ready to receive the data, the CPLD starts to transmit the image header containing the frame index and actual timestamp. Then it waits for valid image data and transmits them untouched. In parallel, it can process the data and calculate image characteristics mentioned above. When all the active pixels in the row are transmitted, the core appends the additional information (row footer) if any and waits for the next sensor action. If it receives another active row data signal, the last two steps are repeated and if it receives the end-of-frame signal it appends the image footer data (calculated image characteristics), generates the end-of-transmission event for the USB chip and returns to the initial wait state. If the USB buffer overflow condition is detected anywhere in the process, it immediately generates the end-of-transmission event and returns to the initial state. On the PC side, the overflow event can be easily detected from the length of the data received in the actual transmission. An example of correctly transmitted data packet structure is depicted in the figure 3.

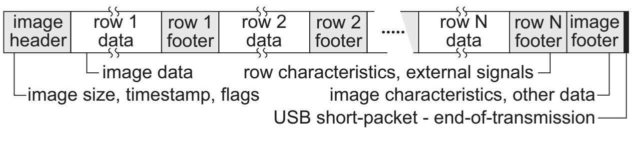

Fig.3. Structure of a single image data packet with additional data encapsulation

Regardless of the measured maximal high-speed USB throughput of around 35 MB/s for a single device connected directly to the USB host controller, the actual data rate may be much lower due to the high latency in the operating system. To verify the CPLD design under different conditions, an universal testbench depicted in figure 4 was designed. It consists of three main blocks/files. The testbench itself, providing the simulation environment, the test data source (the sensor simulator) and the data sink (the USB simulator). All the units are fully configurable and truly simulate the real devices. The virtual sensor can provide any combination of control signals polarity, an arbitrary delay between them and on the data signals it either internally generates simple geometrical objects like bars and rectangles or export a real image read from a file. The USB simulator truly simulates the input buffer management on the Cypress chip, a simplified USB transaction scheme and the latency times caused by data handling performed on PC.

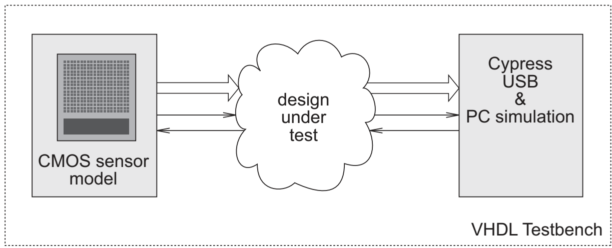

Fig.4. The universal VHDL platform for protocol testing

These latency times were confirmed to be the crucial point of the whole high-speed USB design. With image data rates around 20 MB/s and Cypress chip internal buffer length of 2 KB, the maximum PC latency that still does not cause the buffer overflow is around 100 μs, which is far below the system task manager switching period. The simulations performed with the designed testbench have confirmed the presumption, that these latencies are causing errors observed in the real system. The important part of the simulation is depicted in figure 5.

The plot shows the end of an image scan and the beginning of the next one. In portion (A), the last 4 lines of the image are stored in the buffers and consequently transferred to the PC. With the end of the image, the last packet is transmitted (B), shorter than the previous ones, signaling the PC software, that the image was successfully received and there the kernel driver forwards this information to user-level software together with the received data. Even if the user-level software reacts as fast as possible and provides the driver with a new buffer, the latency in task scheduling is much longer than the vertical synchronization time (C). Even if a simple safety system is implemented and the new image is transferred only if all the input buffers are empty (the empty flag is active), the system has no information, if the PC is ready for new data or not, so it starts to fill the buffers with the data from the next image (D), up to the moment, where all the input buffers are full and an overflow error is generated (E). When the next transaction is initiated by the PC, the data received will be this 'lost' 2 KB followed by the correct data form the next valid image.

Distributed image acquisition model

For interconnection between the CMOS image sensor and user application on the PC, with respect to the real-time data transfer requirements, the distributed image acquisition model was developed. The developed model can be applied on all transfers to PC where transferred data are produced by a fast free-running external source.

While commercial PC cameras provide the acquired image stream over a standardized programming interface (e.g. DirectShow or TWAIN on Microsoft Windows) and all subsequent image processing must be performed in the final application, the distributed image processing model proposed in figure 6 provides the option to preprocess the image before it reaches the target application. The main aim of this approach is to simplify the high-level application and perform all the standard image processing tasks on lower system level where better optimization is possible.

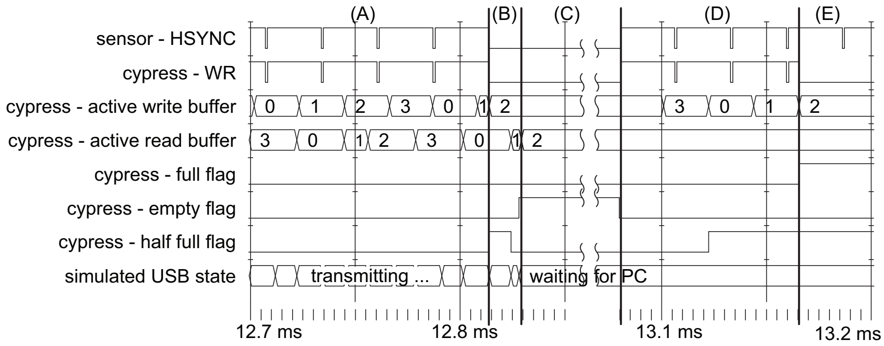

Fig.5. Simulation results: PC-latency-caused error in the high speed image streaming

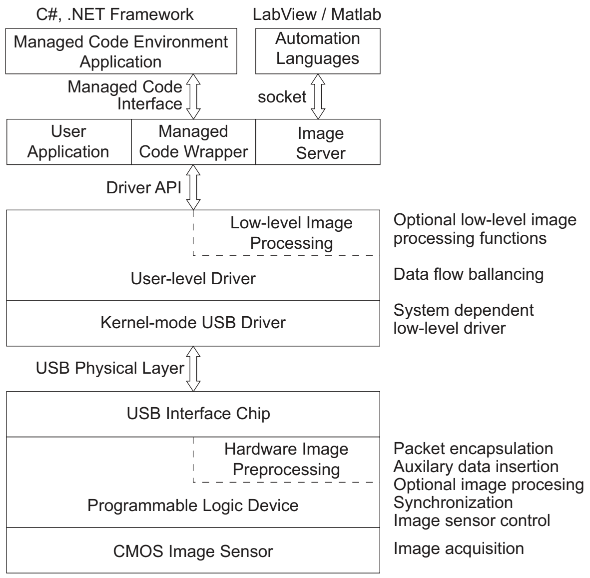

Fig.6. Distributed image acquisition model block diagram

The image processing begins already on the sensor chip. Depending on manufacturer and type, the sensor itself may provide functions for automatic gain and exposition control. The programmable device (CPLD) can then besides the sensor-USB interfacing provide a simple image characterization and preprocessing (thresholding, minimum, maximum intensity calculation), or it can provide a programmable platform for complex custom snapshot control. On the target operating system, the lowest kernel driver usually assures only the data transfer between the physical interface and user-level application. With no additional support, all target applications would have to access the driver, understand the communication protocol and process the data on its own. Instead another layer of abstraction handling the transmission protocol and extracting the acquired images together with all supplementary data is inserted. While the image is received, it can be further processed and additional characteristics may be calculated on the fly. Finally, any user application with compatible programming interface can access this driver and receive preprocessed images with no need for deeper understanding of the underlying protocols. While most of the applications are developed either in managed code environments or advanced graphical or data processing languages (e.g. LabView or Matlab) the lastlayer of universal measurement camera system is a standardized interface for such high-level applications. A simple socket server may provide such interface and distribute the acquired images not only to applications on local computer, but anywhere to the internet.

The system driver

Before describing the final PC-side solution, a short introduction to the PC USB driver field will be given in this chapter. In most common applications, where USB is just a standardized interface for configuration and seldom data transfer, there is no need for concern about speed and optimal system design. In most cases a simple virtual serial port approach (e.g. chips from the FTDI inc.) is sufficient. The problem arises, when a high throughput (above 6 MB/s) is required. Let's assume that there is a hardware solution capable of high speed transfers up to 40 MB/s and focus only on the PC driver part. There are 4 basic approaches.

1. The hardware side is modified to comply with a standard USB class capable of high throughput transfers (e.g. mass storage class or still image capture device class).
2. Hardware manufacturer provides the drivers and communication API.
3. User-space drivers are based on 3rd party universal drivers (e.g. libusb).
4. Customized kernel-mode drivers are developed.

While the first approach gives the maximal compatibility among different operating systems, and simplifies the installation process, it increases the development cost due to the requirements to the USB interface firmware. The last option on the other hand ensures the simplest possible hardware and firmware on the device side, but require a deep knowledge of system driver programming and longer development time. The only acceptable solution would thus be either the dedicated driver created by the chip manufacturer or a 3rd party universal solution. Both approaches were tested with approximately equal preliminary results and the manufacturer-driver-based solution was taken for the final implementation.

In case of Cypress chip, the driver and API consist of two parts. The kernel driver CyUSB.sys and the high-level API distributed as static library containing several classes compiled for Microsoft Visual C and Borland C, whereas both are commercial platforms and there exists no support for open development tools like MinGW32. To support customers with such needs, Cypress also publishes the kernel driver communication protocol (the IOCTL codes and structures) and the whole communication layer of the designed system is based on these functions. But even with this lowest level access, the simple approach of repeating requests for whole images create the latency-caused errors discussed in the previous chapter. The implemented user-level driver solution is depicted in figure 7.

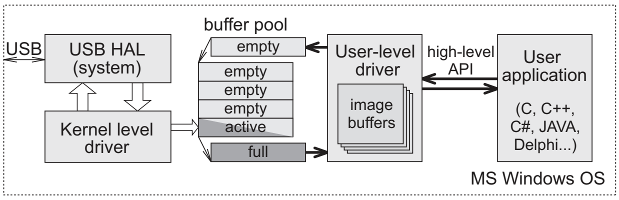

Fig.7. The user-space driver structure with data flow balancing

A middle layer is added to the application – driver communication. This user-level driver only processes the incoming stream and stores the received images and supplementary data from image header and footers into the dedicated image circular buffers, from where they can be fetched by the main application. To minimize the PC latency, the driver keeps a pool of intermediate data buffers and configures the kernel driver in such way, that once one buffer is filled, it automatically takes the next one, without any interaction with the user-level driver. The user-level driver actions are limited by the system scheduler period, so the number of buffers in the pool must be big enough to avoid any data losses in the data stream. If the user application cannot process all received images, the oldest image in the circular image buffer is discarded and replaced by the newest one. When the application fetches the next image, it can test its timestamp or frame index and restore the correct time base.

Besides the simple image acquisition functions, this middle-layer driver provides a set of image processing functions simplifying the common image processing tasks like optimal threshold, mass-center tracking, in-line dimension measurement [12], both with or without the hardware support of the CPLD unit, depending on the camera type.

Conclusion

The developed USB interface for direct interconnection between the CMOS image sensor and PC was introduced in this article. The basic idea, reasonable conception and the final hardware realization of the simple USB digital image acquisition system was presented. As a conceptual model data exchange between PC and fast free-running data sources, the distributed image acquisition model was introduced.

The main advantage of the designed camera solution is the enhanced data stream which consists of both, the acquired uncompressed image data and additional experiment-dependent data. These data can carry an external synchronization information, precise timestamps or actual frame numbering for easy detection of lost frames and image discontinuities. These features destine this camera solution for embedding into laboratory and precise industrial measurement chains. From the image utilization point of view the offered uncompressed image format can decrease uncertainties of image-based measurement results after digital image processing (in position measurement, edge detection, fine textures detection etc.).

For the successful estimation of possible timing problems and maximal USB channel throughput rates a complex VHDL simulation model of image transmission chain was created and its validity confirmed by real-life experiments.

The effective solution of PC-side user level driver was introduced. Having realized a set of practical measurements it has been shown that the achievable continual frame rate from the camera with WVGA (752x480 pixel) CMOS sensor to a standard PC is up to 60 fps (the maximal frame-rate given by the used CMOS image sensor) without any image data lost.

According to the low construction complexity the proposed camera solution is cost-effective and opens the features of expensive image acquisition systems for simpler cost-restricted projects.

Acknowledgements

The described research was supported by research program No.MSM6840770015, sponsored by the Ministry of Education, Youth and Sports of the Czech Republic and by Czech Agency Grant No.GA 102/09/H082.

REFERENCES

[1] Suzuki, Y.; Gai, T.; Yamakawa, M.; Sugiura, H.; , "NTSC/PAL/SECAM digital video decoder with high-precision resamplers,", IEEE Transactions on Consumer Electronics, vol.51, no.1, pp. 287- 294, Feb. 2005, doi: 10.1109/TCE.2005.1405734

[2] KHAN, U. S.; IQBAL, J.; KHAN, M. A.,"Automatic Inspection System Using Machine Vision", In Proceedings of the 34th Applied Imagery and Pattern Recognition Workshop, IEEE Computer Society, pp. 210-217, doi= 10.1109/AIPR.2005.20

[3] Albert J.P. Theuwissen, "CMOS image sensors: State-of-the-art", Solid-State Electronics, Volume 52, Issue 9, Elsevier, ISSN 0038-1101, pp. 1401-1406, doi:10.1016/j.sse.2008.04.012.

[4] Nitta, S.; Tanaka, K.; , "A 79 dB-SNR 70 mW 18 MHz CCD front-end with fully-digital amplification scheme,", IEEE Transactions on Consumer Electronics , vol.47, no.3, pp.459-465, Aug 2001, doi: 10.1109/30.964134

[5] El Gamal, A.; Eltoukhy, H.; , "CMOS image sensors," Circuits and Devices Magazine, IEEE , vol.21, no.3, pp. 6- 20, May-June 2005, doi: 10.1109/MCD.2005.1438751

[6] JUAN, F. MARTIN; LUCIANO CHIANG, "Cost Effective Vision System for Mobile Robots", p. 43, XII International Conference of the Chilean Computer Science Society (SCCC'02), 2002

[7] Abate, L.; Ramponi, G.,"Measurement of the blocking artefact in video frames", 15th IEEE Mediterranian Electromechanical Conference, Malta, 2010, ISBN: 978-1-4244-5794-6, pp. 218 – 223

[8] De Santo, M.; Liguori, C.; Pietrosanto, A., "Uncertainty characterization in image-based measurements: a preliminary discussion", Transactions on Instrumentation and Measurement, Oct 2000, vol.49, no.5, pp.1101-1107, doi: 10.1109/19.872937

[9] Krajčovičová, E.; Martinková, M.; Konopková, R.; Fejtová, M.; Otáhal, J.,"Powerful Software for Image Acquisition and Its Processing", In: Physiological Research vol. 57, num. 4, 2008, p. 80P. ISSN 0862-8408.

[10] Kim, Sanghoon; Willett, J. L.; "An optical microscope for the study of biopolymer solutions under shear field," Review of Scientific Instruments, vol.72, no.11, pp.4252-4260, Nov 2001, doi: 10.1063/1.1412263

[11] Furuta, M.; Nishikawa, Y.; Inoue, T.; Kawahito, S., "A High-Speed, High-Sensitivity Digital CMOS Image Sensor With a Global Shutter and 12-bit Column-Parallel Cyclic A/D Converters,", IEEE Journal of Solid-State Circuits, vol.42, no.4, pp.766-774, April 2007, doi: 10.1109/JSSC.2007.891655

[12] Třeštík, J.; Fischer, J.,"The Optimal Threshold Algorithm and Canny Based Edge Detection in Optical Area and Dimension Measurement", 16th IMEKO TC4 International Symposium, 2008, p. 737-742. ISBN 978-88-903149-3-3.

***Authors**:*
Ing. Ondrej Pribula, E-mail: ondrej.pribula@fel.cvut.cz,
Ing. Jan Šedivý, E-mail: jan.sedivy@fel.cvut.cz,
Doc. Jan Fischer, CSc., E-mail: fischer@fel.cvut.cz
Mail: Department of Measurement, Faculty of Electrical Engineering Czech Technical University in Prague, Technicka 2, 166 27 Prague, Czech Republic