**ISSN**: 1231-4005
**e-ISSN:** 2354-0133
**DOI:** 10.5604/12314005.1216446

# MULTI CAMERA TRIGGERING AND SYNCHRONIZATION ISSUE – CASE STUDY

**Pawel Hyla**

*AGH University of Science and Technology*
*Faculty of Mechanical Engineering and Robotics*
*Kraków, Poland*
*e-mail: hyla@agh.edu.pl*

***Abstract***

*The problem of multi-camera system, in which the synchronization was a priority issue, has been raised in the beginning of the 20$^{th}$ century. It was caused by significant growth of application, in which computer vision technology realized in real-time, mode play the major role. Lately multi-camera synchronization problem is related to three-dimensional reconstruction. It is estimated that 3D imaging market was worth 16.6 billion USD by 2020 year (in 2015 this market was worth only 4.9 billion USD). This constitute raise at 27.64% calculated as CAGR (Compound Annual Growth Rate) indicator. However, presently there exists many issues disturbing in develop full functional 3D imaging systems wherein robust and mapping accuracy are not related with the system total price. Multi-camera imaging (MCI) technology is a perfect candidate to obtain 3D imaging, moreover the systems contains this type of solution already existed but they possess limitation. First of all, MCI are perfect for reconstruction static objects.*

*This paper describes the common known problem concerning multi-camera system in which correlation between independently taken images from the multiple viewpoints must be extremely high in the time domain. However, some kinds of application do not require perfect snapshot synchronization but time delay must be exactly known. Generally, camera synchronicity issue can be achieved through hardware or software solution. Hardware triggering usually ensures high synchronicity precision and is robust but it is always associated with expensiveness. In turn of software triggering the application architecture in hardware point of view are simpler and inexpensive although they are uncertainty as hardware solution. Additionally, in the paper author's main attention was focused on possibility of synchronization pentadruple cameras system with GigE interface with hardware and software triggering and estimation software solution average time delay in comparison with hardware triggering.*

***Keywords****: multi-camera synchronization, GigE, hardware triggering, software triggering*

## 1. Introduction

Vision is the one of most exiting sensing mechanism capable to use on very large scale [5]. Many biological systems use vision formed in the cycle of millions years of natural evolution process and use it as their most prominent way for gathering information about their environment. Vision is a perfect candidate for gathering information about surrounding environment for machines and robots [3, 5, 11].

The camera as a tool in science can deliver a lot of information about surrounding world [13] and helps to understand the many physical phenomena's. However, today the major camera role has changed. The vision devices are not only used to object and process observe but on the vision data, digital mapping and object representation must be achieved. The problem of multi-camera system [2, 9, 12], in which the synchronization was a priority parameter has been raised in the beginning of the 20$^{th}$ century. It was caused by significant growth of application in which computer vision technology realized in real-time mode [7] play the major role. However, presently there exist many issues disturbing in develop full functional 3D imaging systems wherein robust and mapping accuracy are not related with the system total price. Multi-camera imaging (MCI) [1] technology is a perfect candidate to obtain relatively cheap 3D imaging [4, 6], moreover the systems contain this type of solution has already existed but they possess a lot of limitation [6].

This paper describes the common known problem concerning multi-camera system in which correlation between independently taken images from the multiple viewpoints must be extremely high in the time domain. However, some kinds of application do not require perfect snapshot synchronization but the time delay must be exactly known. Generally, camera synchronicity issue can be achieved through hardware or software solution. Hardware triggering usually ensures high synchronicity precision and are robust, but it always associates with expensiveness and raises the total cost. In turn of software triggering the application architecture in hardware point of view are simpler and inexpensive although they are uncertainty as hardware solution. Additionally in the paper author main attention was focused of possibility synchronization pentadruple cameras system with GigE interface with hardware and software triggering and estimation software solution average time delay in comparison with hardware triggering.

## 2. Vision system for hardware and software trigger delay measure – an architecture

As a test platform, an industrial PC built in PXI (*PCI eXtensions for Instrumentation*) standard was used. The PXI B (Fig. 1.) main controller it was equipped with 2.3 GHz eight-core Intel Xeon processor and 24 GB of RAM (*Random Access Memory*) memory. The computing power of the described unit measured in FLOPS (*FLoating Operations Per Second*) were 271 Gigaflops at a CPU (*Central Processor Unit*) frequency 2.2945 GHz. Applied PXI architecture allows to use three independent NIC (*Network Interface Card*) with dual GigE board each. To obtain hardware trigger functionality another PC in PXI standard was used to generate external voltage trigger signal (PXI A). On the PXI the real-time OS was deployed. The entire architecture of the vision system was presented in Fig. 1.

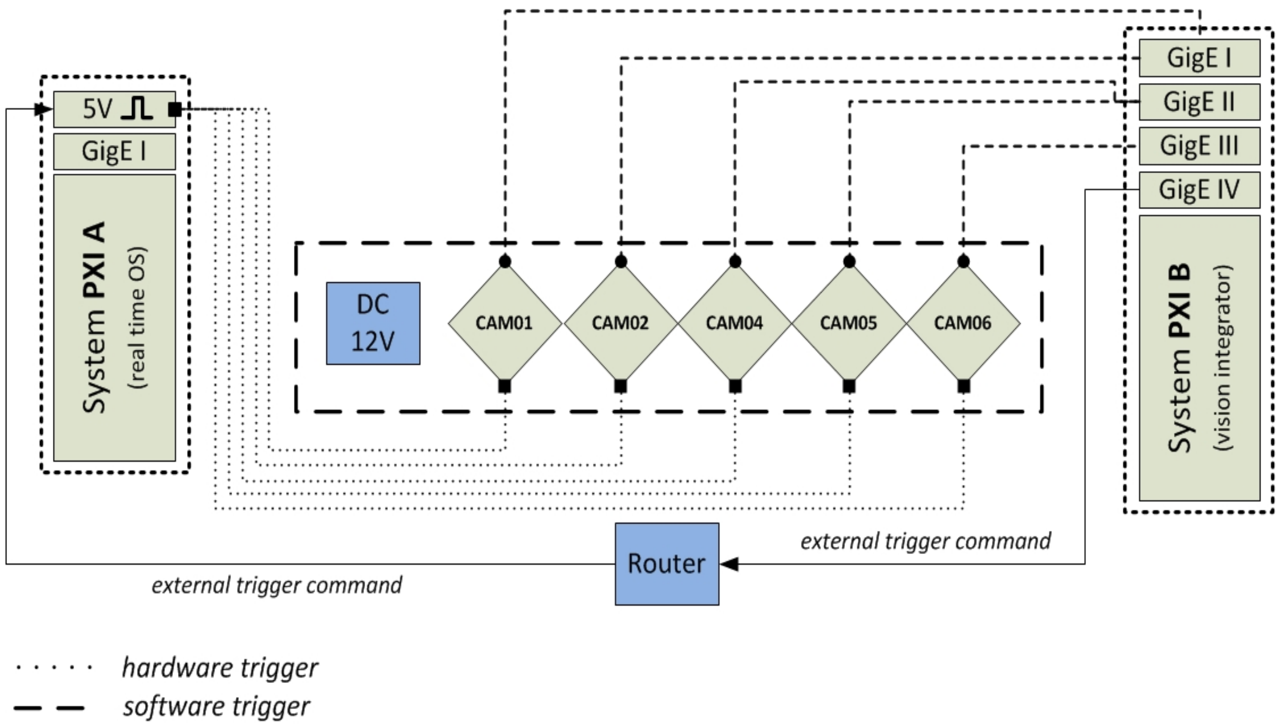


*Fig. 1. The architecture to estimate hardware vs. software trigger time delay*

For the experiment purposes the cameras array (see Fig. 2.) was built. The array contains five Basler SCA640-70GM unit with GigE interface. A maximum cameras resolution (659x490) to the test was used, additionally the exposure time was as short as possible with the subject to the each caught image must be enough bright. With use of a special holder each cameras optical axis was spaced on 65 mm distance in the perpendicular directions.

*Fig. 2. The pentadruple GigE camera array – front view*

Relatively small camera optic axis displacement allows capturing the same visible area in each device. In the camera field of view two-computer monitors display different content were placed. The first used monitor was a CRT (*Cathode-Ray Tube*) type; the second was a well-known LCD (*Liquid Cristal Display*) display. On the first one, there was display a pattern that allows observing a horizontal refresh rate while the second show a counter with 0.01s accuracy (see Fig. 3).

*Fig. 3. From the left: CRT (pattern) and LCD (counter) monitors are visible*

## 3. Physical principles of measurement

To generate the image on the CRT monitor an electron beam produced at the cathode ray is accelerated and focused to strike on the screen covered by phosphor layer. In a CRT monitor it is possible to distinguish two basic parameters: horizontal and vertical refresh rate. Horizontal refresh rate determines how many times per second the electron beam travels along a single horizontal line and returns to the beginning. This frequency is expressed by kilohertz (kHz). The number of horizontal lines depends on the monitor screen resolution. Whereas vertical refresh rate specifies how many times per second the entire screen was refreshed. However, the CRT monitor refresh rate is the number of horizontal lines multiplied by the vertical frequency.

To measure the camera unsync time with CRT monitor help is necessary to capture the same screen by the all tested devices. Afterwards, by analyse of the corresponding frames taken from the cameras. In our case the first, second, third, fourth and fifth cameras frames must be caught simultaneously to determine unsync time. In each recorded frame it is possible to see the position of the CRT monitor's electron beam at the display pattern. The electron beam positions will be different for the same frames caught by others camera when they are unsynchronized. Depending on the camera exposure time, more than one line of electron beam can be visible in each frame. With known vertical and horizontal refresh rates it possible to calculate the time in which the electron beam draw one line $t_{sl}$ (1) and consequently time need to draw an entire single frame $t_{sf}$ on the display (2):

$$t_{sl} = \frac{1}{f_H}, \tag{1}$$

$$t_{sf} = \frac{1}{f_V}, \tag{2}$$

where:
$f_H$ – horizontal refresh rate,
$f_V$ – vertical refresh rate.

To calculate the total unsync time, in case when we examine more than two cameras it is necessary to choose the reference device. Usually a reference device is the camera, which caught the earliest frame. The total unsync time can be calculated only in comparisons with the reference device frame, so in fact it allows to formulate equations to calculate the total unsync time, presented bellow (3):

$$t_{\Delta t} = \frac{x_2 - x_1}{f_H} + \frac{y_2 - y_1}{f_V}, \tag{3}$$

where:
$x_1$, $x_2$ – electron beam position on the pattern in each caught frames,
$y_1$, $y_2$ – duration of the one screen refresh cycle in each caught frames.

### 3.1 Trigger action

#### 3.1.1. Hardware trigger

Generally, some type of measurements can be realized any time and it is not relevant when the start or stop was realized. Usually, the static measurements do not need the triggering [1]. Unfortunately, the static measurements in vision application are rather than any form of measurements of time varying phenomena's. Additionally the camera constitutes a natural equivalent and expansion human eye capabilities and it has usually a much greater technical capabilities in surrounding world observation. Hardware triggering allows executing an action to begin when some definite condition is met [8]. To induce hardware trigger the right digital electronic signal is necessary. In cameras, there is usually some type of voltage signal. The data collection start/ stop signal was induced through specific criteria will be met. This kind of hardware control of the any device type and the measurement process is called hardware triggering. In the theory, it is simple, but in real measurement process vision hardware, triggering is a complicated issue. In some experiments to obtain a proper signal to induce, trigger action an analog signal must be converted into digital form, which might be awkward. The hardware trigger effect on pentadruple camera system was presented in the Fig. 4.

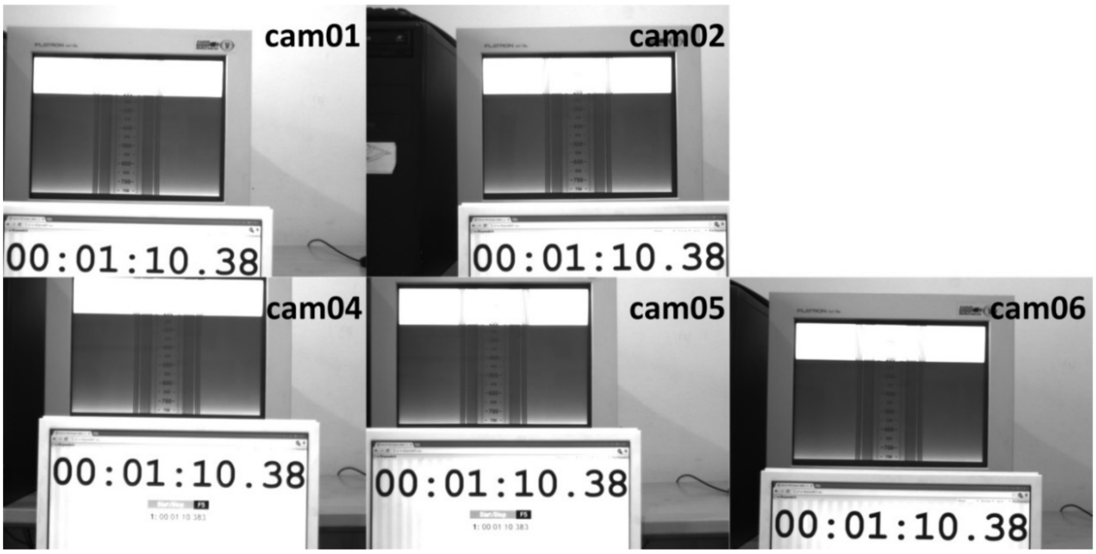


*Fig. 4. External hardware trigger images assembly from five cameras*

As it can be observed, each snapshot was taken in the same period. The counter on the LCD screen shown the same value (00:01:10.38), additionally the electron beam on the CRT was caught in the same place (approximately on the 200 position at the pattern). However, simultaneously activation of the camera shutter under the hardware trigger action does not raise amazement but constitutes a reference point to another software trigger experiment.

### 3.1.2. Software trigger

To determinate when the measuring process start/stop is possible too through the proper configuration a type of software code to command it. This is the main idea behind a "software triggering". Software triggering involves two types of issues. The first one involves the "chicken" or "egg" paradox. We do not want the measurements captured until we trigger but on the other hand we do not have any reference point in triggering. The second type of problem is constituted with triggering many devices in the same period using software command. In case of set cameras participating in some type of vision system – the shutter time delay of each camera is not exactly known and that constitutes the basic issue.

## 4. Multi-camera system – software trigger

The architecture of the tested system was presented and described in the chapter 2. To the test of the multi-camera system releasing with software trigger command it was used array containing five Basler SCA640-70GM camera with GigE interface. Three independent series of measurement was conducted (Fig. 5.) and on this base the software trigger calculation was done (Fig. 6-8).

The main difficulty in the presented data interpretation depends on the applied method disadvantage. In the each measure test the cameras unsync must be calculated in reference to the fastest device. It impossible to predict the fastest device in each test because in each step different device can be fastest. The described problem is visible in Fig. 6-8, where accordingly the fifth (Fig. 6), fourth (Fig. 7) and again the fifth (Fig. 8) devices were the fastest, therefore the time delay for this devices were accepted as zero second. The counter on the LCD screen on all taken snapshots (within each test set) shows the same value adequately 0:00:07.23 in first row, 00:04:29.56 in second row and 00:07:42.65 in the third row in the Fig. 5. This indicates that difference in time delay for all taken frames is under $1.0 \times 10^{-2}$ s. After calculation in applied architecture, the average software trigger delay was reached $1.1 \times 10^{-4}$ s., calculated as value relative to the fastest camera in the each experiment.

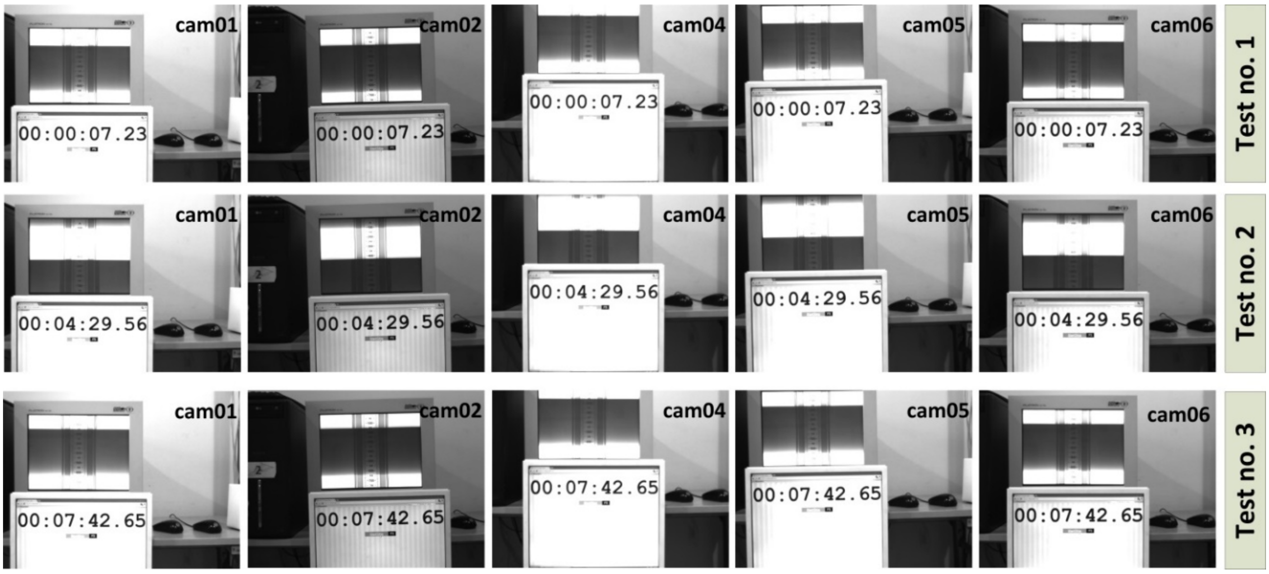


*Fig. 5. Software trigger actions*

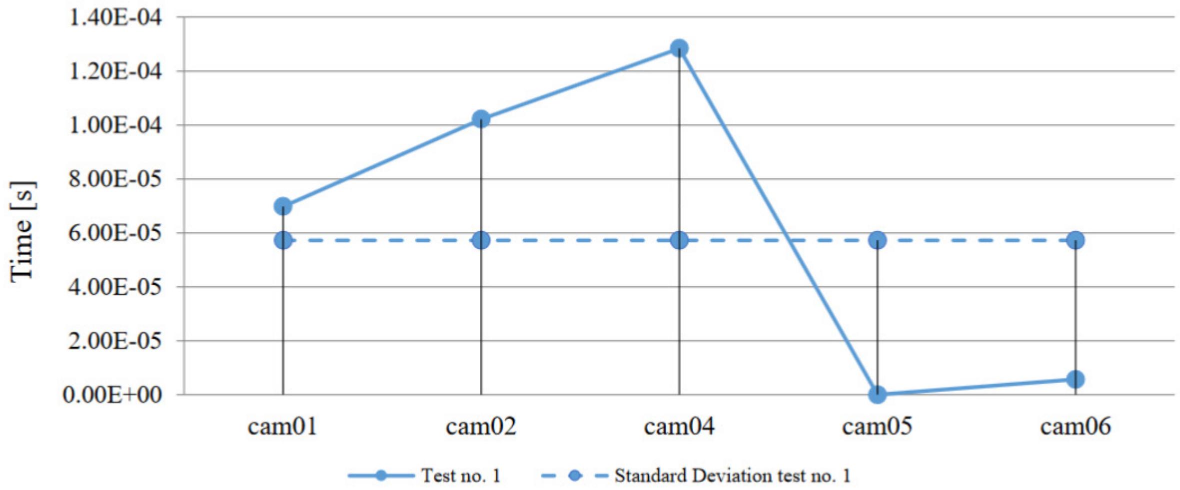


*Fig. 6. The software trigger (solid line) time delay in reference to camera 5 with calculated standard deviation (dashed line)*

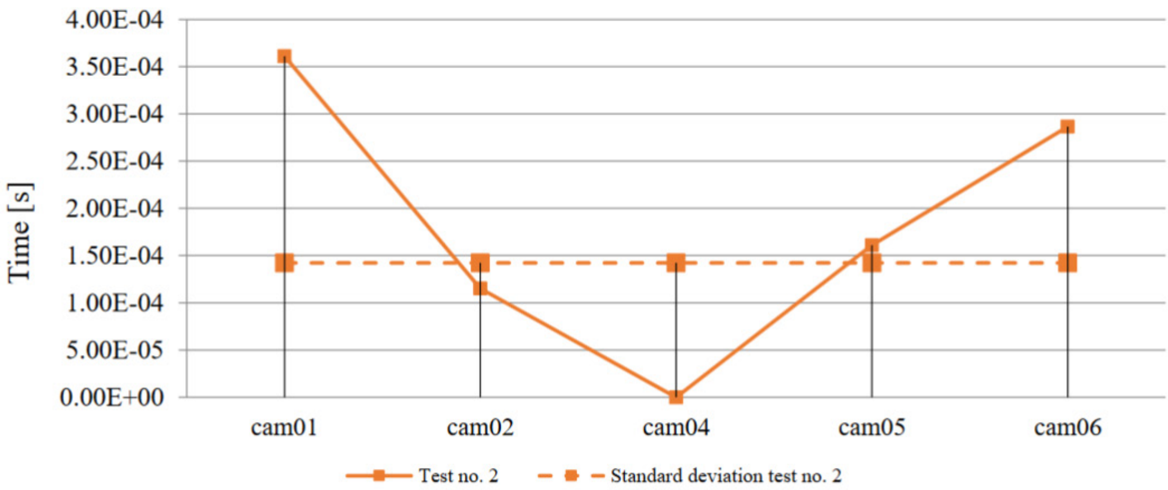


*Fig. 7. The software trigger (solid line) time delay in reference to camera 4 with calculated standard deviation (dashed line)*

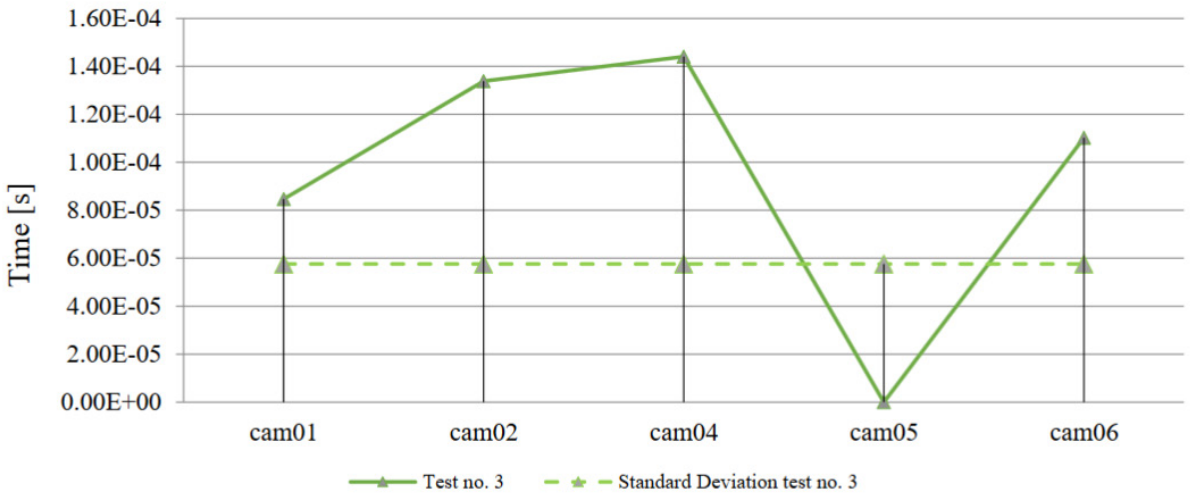


*Fig. 8. The software trigger (solid line) time delay in reference to camera 5 with calculated standard deviation (dashed line)*

## 5. Final remarks

The obtained results allow concluding that it is possible to build and configure multi-camera vision software trigger system and achieve results similar to hardware solution. However, the GigE interface is good choice for multi-cameras system with particular focus any trigger solution. The presented interface determines several conditions that must be met at the same time to achieve hardware trigger functionality on software application, most of all:

- the CPU of the used vision terminal must have enough computer power,
- each camera must be connected to the own GigE expansion card and the all infrastructure must be "jumbo frame" compatible,
- during the all test pentadruple camera array Ethernet cable linked cameras with GigE adapter do not exceed 5 m, therefore the longest cable can change the final results,
- each camera was connected to the same stabilized DC power source.

It should be emphasized that to obtain the listed condition in any infrastructure is not cheap, therefore the total prize of the software and hardware trigger solution cannot be compared. However at the start of experiment the camera array contain six devices but under the test the *cam03* was death after receiving on of the external trigger signal. It may suggest that hardware trigger in some cases can be deathly; therefore the software solution is much safer for the devices.

**Acknowledgment**

*The research project is financed from the Polish Science budget in the year 2016.*

**References**

[1] Aghajn, H., Cavallaro, A., *Multi-Camera Networks Principles and Applications*, Elselvier Inc., USA 2009.
[2] Bajramovic, F., *Self-Calibration of Multi-Camera Systems*, Berlin 2010.
[3] Faugeras, O., *Three-Dimensional Computer Vision A geometric viewpoint,* MIT Press, London 1993.
[4] Hyla, P., *Stereovision system for overhead travelling crane workspace visualization – validation approach,* Proceedings of the 18th International Conference *of* Methods and Models in Automation and Robotics MMAR 2013, pp. 69-74, Międzyzdroje 2013.
[5] Hyla, P., Szpytko, J., *The application of image analysis methods in selected issue solution dedicated for overhead travelling crane*, Journal of KONES Powertrain and Transport Means, Vol. 21, No. 2, pp. 97-104, Warsaw 2014.

[6] Kang, S. B., Szeliski, R., Chai, J., *Handling occlusions in dense multi-view stereo*, in Proc. CVPR 2001, pp. 1-8, 2001.

[7] Rai, P. K., Tiwari, K., Guha, P., Mukerjee, A., *A Cost-effective Multiple Camera Vision System using FireWire Cameras and Software Synchronization*, 10th International Conference on High Performance Computing (HiPC 2003), Hyderabad, India 2003.

[8] Lee, L., Romano, R., Stein, G., *Monitoring activities from multiple video streams: Establishing a common coordinate frame*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 8, pp. 758-767, 2000.

[9] Moore, C., Duckwortht, T., Aspin, R., Robert, D., *Synchronization of Images from Multiple Cameras to Reconstruct a Moving Human*, Distributed Simulation and Real Time Applications (DS-RT), IEEE/ACM 14th International Symposium, pp. 53-60, 2010.

[10] Sinha, S., Pollefeys, M, *Synchronization and calibration of camera networks from silhouettes*, In Proc. of ICPR, pp. 116-119, 2004.

[11] Szpytko, J., Hyla, P., *Disparity compute methods in three-dimensional scene reconstruction for overhead travelling crane work space visualization,* Journal of KONES Powertrain and Transport Means, Vol. 19, No. 3, pp. 421-428, Warsaw 2012.

[12] Thiel, D., Goulart, M., Botelho, S., Bem, R., *Hardware and Software Infrastructure to Image Acquisition using Multiple Cameras,* Universidad Federal do Rio Grande – FURG, 2004.

[13] Wilburn, B., Joshi, N., Vaish, V., Talvala, E. V., Antunez, E., Barth, A., Adams, A., Horowitz, M., Levoy, M. *High performance imaging using large camera arrays*, ACM Trans. Graph., Vol. 24, No. 3, pp. 765-776, 2005.