

WIRED AND WIRELESS CAMERA TRIGGERING WITH ARDUINO

H. Kauhanen^{a,*}, P. Rönholm^a

^a Aalto University, School of Engineering, Department of Built Environment, P.O. Box 14100, 00076 AALTO, Finland
(heikki.kauhanen, petri.ronholm)^a@aalto.fi

Commission I, WG I/5

KEY WORDS: Camera, Triggering, Wired, Wireless, Arduino

ABSTRACT:

Synchronous triggering is an important task that allows simultaneous data capture from multiple cameras. Accurate synchronization enables 3D measurements of moving objects or from a moving platform. In this paper, we describe one wired and four wireless variations of Arduino-based low-cost remote trigger systems designed to provide a synchronous trigger signal for industrial cameras. Our wireless systems utilize 315MHz or 434MHz frequencies with noise filtering capacitors. In order to validate the synchronization accuracy, we developed a prototype of a rotating trigger detection system (named RoTriDeS). This system is suitable to detect the triggering accuracy of global shutter cameras. As a result, the wired system indicated an 8.91 μ s mean triggering time difference between two cameras. Corresponding mean values for the four wireless triggering systems varied between 7.92 and 9.42 μ s. Presented values include both camera-based and trigger-based desynchronization. Arduino-based triggering systems appeared to be feasible, and they have the potential to be extended to more complicated triggering systems.

1. INTRODUCTION

Triggering cameras in multi-sensor systems enables simultaneous data capture from several cameras. An accurate synchronization of cameras allows for 3D measurements of objects from moving or shaking platforms. In addition, objects can be moving in the area of interest. From the application point of view, accurate synchronization enables many tasks that are difficult to make with other measuring systems.

Triggering alternatives include software-based (Litos et al., 2006) and hardware-based (Thiel et al., 2013) methods. The hardware-based method is more accurate. Industrial cameras often offer a precise hardware trigger port that can be utilized to obtain synchronous operation. However, devices to generate a triggering signal are often expensive. Our previous low-cost wired triggering system was based on the LPT port of a desktop computer. An LPT port has eight data pins; each pin can control an industrial camera or a group of cameras. Because the LPT port is usually not available in modern computers, we were motivated to develop a system that is able to achieve the same or better triggering capabilities without a desktop computer and LPT port. An Arduino Uno microcontroller was chosen for this task.

Arduino is a prototyping board designed to allow people, such as artists and hobbyist, to implement their own electronic device prototypes without much knowledge about electronic engineering. Arduino uses a very simple and well-documented programming interface and a wide variety of add-on packages, ranging from robotic automation to environmental monitoring. Most of these components are readily available and easy to implement into an Arduino prototype project. Therefore, Arduino allows simple, effective and low-cost prototyping. With proper instructions, Arduino prototypes can be implemented by anyone with simple soldering skills, which makes it an interesting development environment.

The aim of this paper is to describe a couple of variations of low-cost remote trigger systems designed to provide a synchronous trigger signal for industrial cameras. Our triggering device prototypes utilize readily available parts such as Arduino microcontrollers and radio transmitter/receiver pairs often found in car remote keys.

2. METHODOLOGY

Our wired Arduino prototype has 14 pins that can be utilized to trigger industrial cameras or camera groups. It is also possible to trigger several cameras with each of the pins. Commercial cameras can be triggered as well by changing the firmware of the triggering device. This is needed because different camera manufacturers use different kinds of signals to trigger their cameras. However, this was not experimented with in this paper. Commercial cameras are usually triggered by shorting the trigger pin to ground, while industrial cameras are designed to trigger when they sense a certain voltage across the trigger pin and ground. This means that most industrial cameras can be triggered directly from the Arduino data pins.

The wired Arduino remote triggering system was also extended to be able to wirelessly trigger cameras and other sensors. A wireless solution is especially interesting because it enables easier and versatile placement of cameras in the space. In the most advanced solution, each camera or camera group is attached to a battery-powered nanocomputer handling data storage, and the cameras are synchronized through a wireless system. Such an installation saves a lot of time when a camera network is assembled in a new environment. It also gives flexibility by allowing the optimal placement of cameras, even in difficult cases where using wires would be difficult or impossible. For example, if there are moving objects that cannot be stopped between cameras, it can be difficult or even dangerous to spread wires around for triggering cameras. In

* Corresponding author

addition, only the range of the Arduino radio signal limits the distance between cameras.

The wireless system was implemented by applying a radio transmitter/receiver pair for each camera or camera group that is triggered using one of the Arduino data pins. Our prototype has an option to use either a 315MHz frequency or a 434MHz frequency. In addition, a dual frequency operation is also possible. This alternative is useful if there are two camera clusters that take images with different frequencies, but still synchronously. Our current wireless prototype version triggers the camera almost directly from the signal pin of the receiver with only minor analog filtering. However, some filtering is needed to separate the signal from background noise, especially in noisy conditions, when there is other traffic in the 315MHz or 434MHz bands. Because of this, we use a simple capacitor that which filters most of the noise from the radio signal.

The principle of using a capacitor to filter the noise is that when the signal output pin of the RF receiver outputs a voltage, it is charging a capacitor. The capacitor is connected between the signal output pin and the ground pin of the RF receiver. When the capacitor has been charged enough, it will trigger the camera. If the signal includes only noise, it will take a relatively long time to charge the capacitor. If the capacitor is sized correctly, the cumulative noise between the image acquisitions is not strong enough to charge the capacitor, and thus the camera will not trigger inadvertently. Such a simple analog approach introduces only very small latency to the synchronization signal, which is important. There are no digital processors in the signal path from an Arduino data pin to the camera trigger port, which assumably leads to good synchronization accuracy.

3. TRIGGERING SYSTEM SETUPS

The simplest form of our triggering prototype is the wired Arduino trigger unit as shown in Figure 1. The system consists of only an Arduino Uno microcontroller board and two wires. One wire is the trigger ground and the other wire is the trigger signal. The signal cable can be attached to any of the Arduino Uno 14 data pins. The data pins can be configured to act as outputs or inputs. In all our examples, we configure data pins as outputs. Arduino Uno can be powered from a USB input or by an external power supply with a voltage range of 6V to 20V. The recommended voltage range is 7V to 12V.

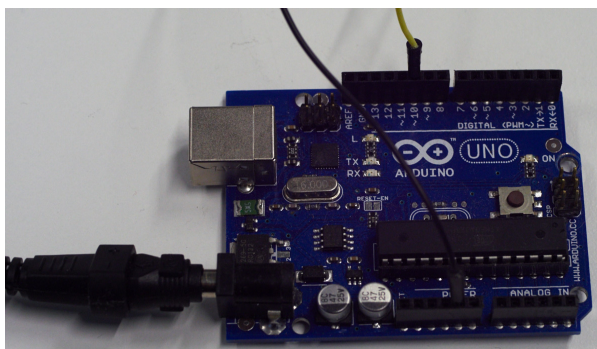


Figure 1. The Arduino Uno microcontroller setup for wired camera triggering. The black wire goes to the ground pin, and the yellow wire gives the trigger signal from the Arduino Uno data output pin.

For wireless triggering, we use an Arduino data pin to operate a WRL-10535 Radio Frequency (RF) transmitter, which sends the trigger signal by an analog radio transmission to the WRL-10535 RF receiver. The transmitter is powered from the 5V output of the Arduino Uno board, as shown in Figure 2 (the red wire). It is recommended to supply at least 7V to the Uno board to make sure the 5V output from the board is stable. The yellow wire is the trigger signal and the white wire is the ground.

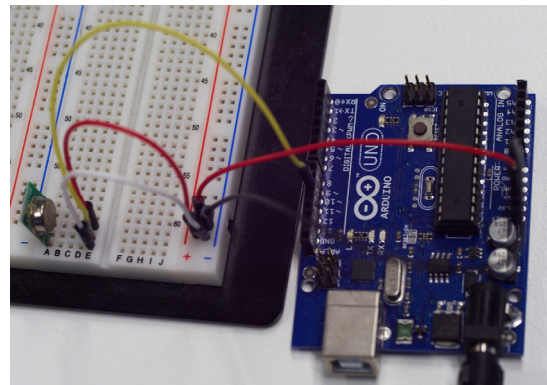


Figure 2. The Arduino Uno wireless trigger setup. The red wire is a 5V voltage wire that powers the RF transmitter, the white wire is for the ground, and the yellow wire gets the trigger signal from the Uno data output pin.

Figure 3 shows two WRL-10533 RF receivers. They receive the signal sent by the WRL-10535 transmitter, and the data pin can be used to trigger the camera. We recommend using a film box polyester capacitor to filter the receiver data pin noise as shown in Figure 3. Figure 4 shows the simple capacitor filter we applied to reduce noise of the RF receiver data output pin. The use of the filter capacitor is discussed in more detail in the Discussion chapter.

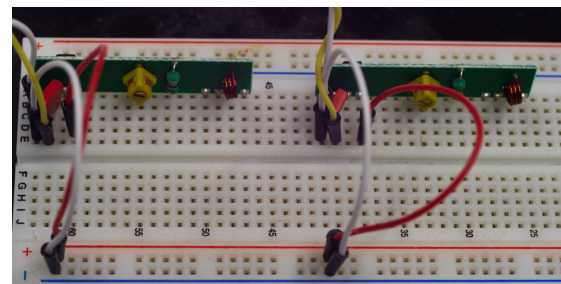


Figure 3. Two RF receivers with 0.068 μF filter capacitors. Receivers are powered by a 5V voltage source via red wires. White wires are for the ground and yellow wires are the camera trigger wires.

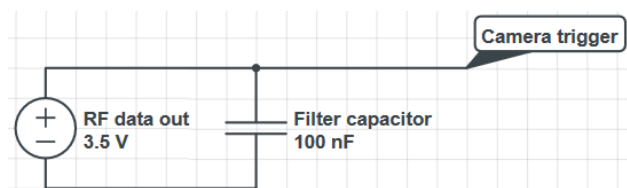


Figure 4. The schematic of the RF receiver filter capacitor. A dual frequency operation using two different frequency transmitter/receiver pairs was also tested. Figure 5 shows the

dual frequency transmitter setup. Note that in this case, the yellow signal wire is split to both the transmitters, but two data output pins can be used if two differently timed trigger signals are desired.

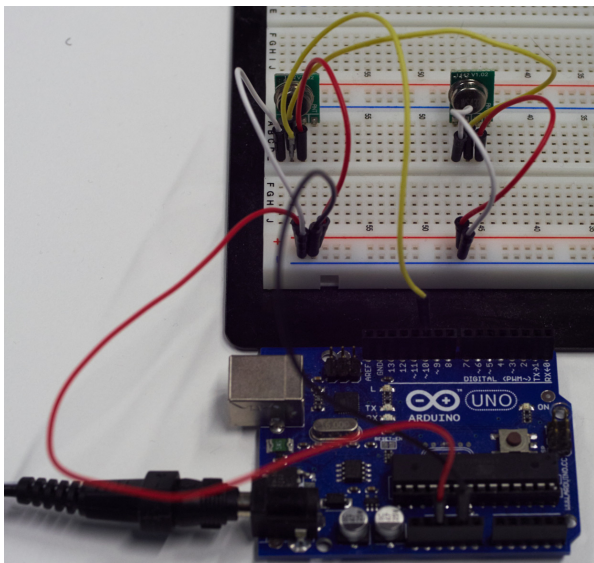


Figure 5. Dual frequency wireless setup using 315MHz and 434MHz transmitters.

4. TRIGGER DETECTION

In order to measure triggering delay between cameras, a prototype of a rotating trigger detection system (named RoTriDeS) was developed (Figure 6). The system utilizes a modified cooling fan. A clock pointer was attached to the fan motor housing. To assist automatic measurements, three red targets were placed in the system. One target is at the rotation center of the system, the second target is on the moving end of the pointer, and the third target is assembled onto the frame, which is not moving.

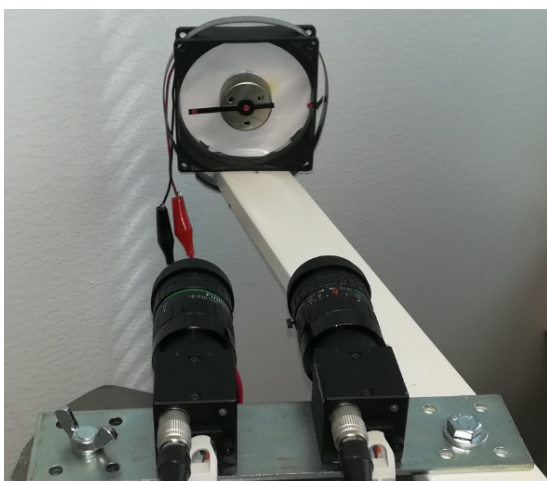


Figure 6. The rotating trigger detection system for two cameras.

Cameras were mounted in such a way that the image planes were close to parallel to the rotation plane of the pointer. In addition, the perpendicular distance from both cameras to the

rotation plane were closely the same. The rotation speed is dependent on how much voltage is selected. We utilized an adjustable external power source that was set to 14.4 V. In our case, this corresponded to 104 rotations per second (6240 RPM), which was measured with a Turnigy Tachometer. In other words, the difference of one degree equals to a 26.71 microseconds time difference.

In this experiment, two identical industrial cameras were utilized. Point Grey's GigE global shutter cameras BFLY-PGE-50S5C-C with Fujinon HF-818-12M lenses were set to capture full 2448 x 2048 color images. To capture the moving pointer with a minimum blur, the shortest 0.022 ms exposure time was applied. In Figure 7, an example of simultaneously captured images is presented.

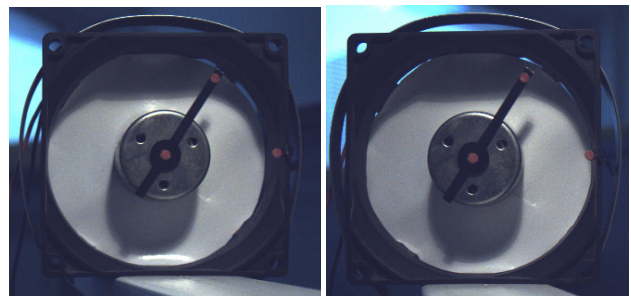


Figure 7. An example of simultaneously captured images with two triggered cameras.

The center points of the red targets were measured automatically. However, finding the seed points of the targets was the first step. The rotation center and the static target on the frame needed manually pointed initial locations as seed points, but the moving target required a more advanced system. First of all, a limited search space was manually selected. Then, the purest red pixel was searched and considered as a seed point.

In the next step, the center points of targets were computed. For this, a region growing algorithm was utilized starting from the seed points. Because the target plane and both image planes were close to parallel, it was assumed that the effect of the target eccentricity due to the projective geometry (Luhmann, 2014) can be neglected. Also, a possible effect of chromatic aberration of color bands (Cronk et al., 2006) was neglected. In practice, the region growing algorithm utilized only red and green bands. Because the lighting conditions were not changing during the image acquisition, simple thresholds for region growing were set empirically. These thresholds included the minimum acceptable value for red color and another threshold for the ratio of red and green bands. In order to prevent possible division by zero, both color values were increased by one. A simple centroid was computed from the pixels belonging to a target. This method has been reported to give a good sub-pixel accuracy (Shortis et al., 1995).

Two lines were computed for each image to reveal the angle of the pointer (Figure 8). The first line passes through the rotation center of the pointer and the moving point target of the pointer. The second line goes through the rotation center of the pointer and the static target on the frame. The angle between these two lines was computed and compared with the corresponding angle from another image. Such angular differences were converted into microseconds describing the triggering delay between cameras.

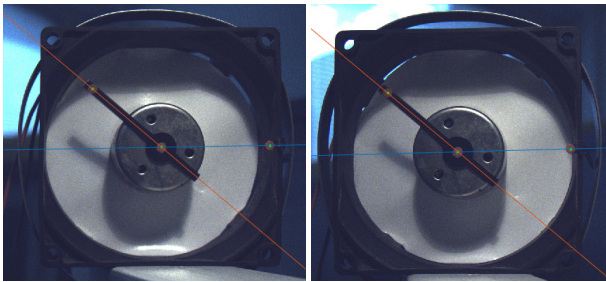


Figure 8. The angles between found lines were measured and compared between images from two different cameras. In this case, the angle difference was 0.287 degrees, which indicates a 7.7 μ s triggering difference.

5. RESULTS

As the reference, the results of the old LPT-based wired triggering system were included in the examination. The triggering time differences between cameras in the wired system are illustrated in images 9 and 10. In all experiments, 100 images with both cameras were taken with the same settings.

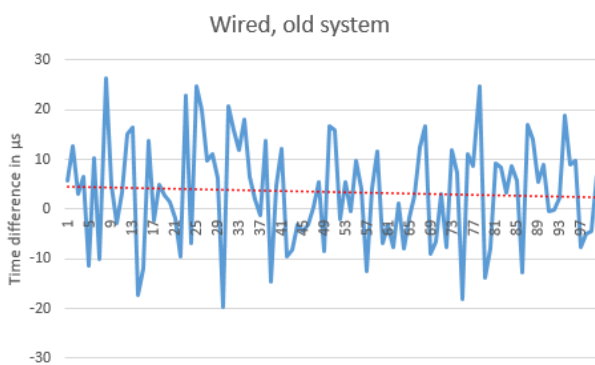


Figure 9. Triggering differences for 100 samples (blue line) and a trend line (red line).

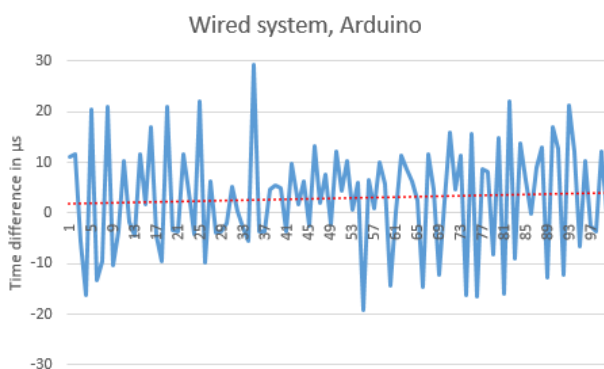


Figure 10. Triggering differences for 100 samples (blue line) and a trend line (red line).

The behavior of noise in the case of wireless triggering was examined separately in a normal household environment, which included typical radio noise such as cell phones, radios, and Wi-Fi base stations. During this test, the distance between wireless

transmitter and receiver was 10 m, and there was a mineral wool insulated drywall between the RF transmitter and two RF receivers. Figure 11 shows the trigger signals measured with a digital DSIM12 oscilloscope from the RF receiver data output pins after the capacitor filter. The graph shows strong noise before the RF transmission starts, but as soon as the transmitter starts to transmit, the signal becomes clear and shows very low noise levels and good synchronization accuracy. Most industrial cameras can be set to trigger from a rising or falling edge. Our signal uses an equal time for the “high” and “low” states, thus resulting in the same frequency for the image capture regardless of whether the camera triggers on the rising or falling edge of the signal.

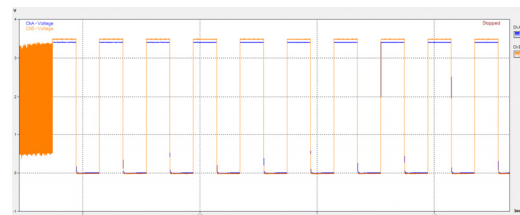


Figure 11. An oscilloscope measurement series of the two RF receiver data output pins.

Figures 12 to 15 illustrate the triggering time differences in the wireless triggering systems. The main results are collected in Table 1, which presents the mean, standard deviation, minimum, and maximum values of the triggering time differences. Again, each test case included 100 samples.

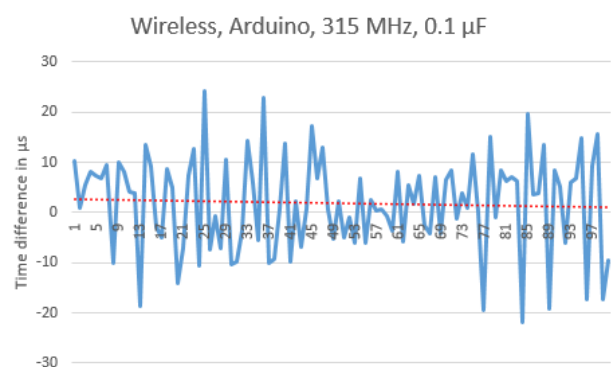


Figure 12. Triggering differences for 100 samples (blue line) and a trend line (red line).

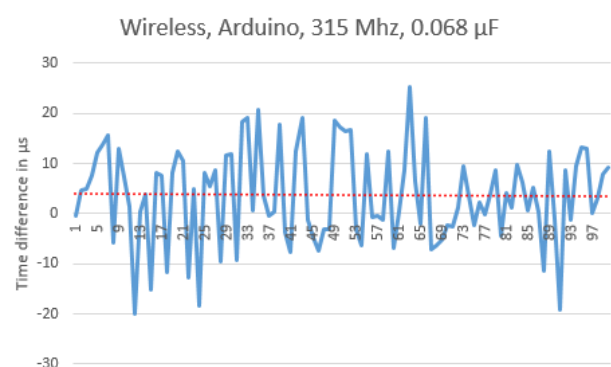


Figure 13. Triggering differences for 100 samples (blue line) and a trend line (red line).

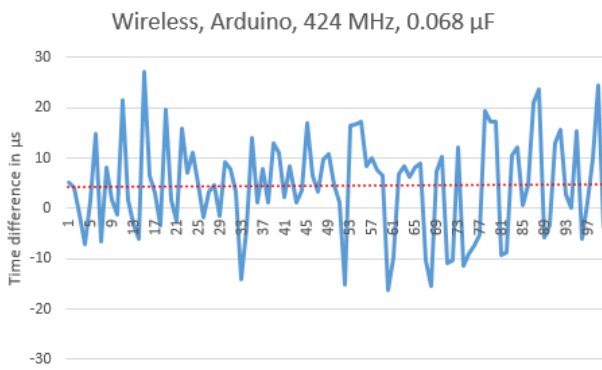


Figure 14. Triggering differences for 100 samples (blue line) and a trend line (red line).

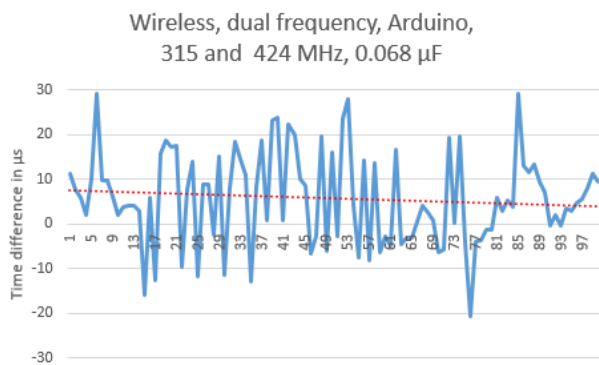


Figure 15. Triggering differences for 100 samples (blue line) and a trend line (red line).

Table 1. Statistics for all experiments. The number of samples was 100 in each case.

| | Mean (μs) | STD (μs) | min (μs) | max (μs) |
|---|-----------|----------|----------|----------|
| Wired, old system | 9.10 | 6.11 | 0.15 | 26.42 |
| Wired, Arduino | 8.91 | 6.10 | 0.01 | 29.31 |
| Wireless, 315 MHz, 0.1 μF | 7.92 | 5.52 | 0.07 | 24.33 |
| Wireless, 315 MHz, 0.068 μF | 8.01 | 6.08 | 0.16 | 25.41 |
| Wireless, 424 MHz, 0.068 μF | 8.72 | 6.09 | 0.02 | 27.08 |
| Wireless, dual, 315 and 424 MHz, 0.068 μF | 9.42 | 7.10 | 0.32 | 29.28 |

6. DISCUSSION

Because of noise, our approach might lead to malfunctions if the filtering capacitor is saturated by the noise. The sources of noise include the noise of the RF transmitter, the noise of the RF receiver, and the RF noise of the operating environment. Especially the RF noise of the environment can be very high in many situations since both 315MHz and 434MHz bands are widely used by many communication devices, such as remote car keys. The results in Figure 11, however, show that in normal cases no extra triggers occur in normal conditions. If disturbing signals become an issue, it is possible to program the

Arduino to use more complex functions by using a so-called handshake trigger signal pattern that would then be received by another Arduino microcontroller attached to the radio receiver. The receiving Arduino board would then use one of the data pins to trigger the camera via a wired signal. However, this vision was not implemented in this research.

Because of the analog nature of our prototype, we are limited to using only certain intervals of camera triggering. If we pause for too long between the trigger signals, the camera will eventually start to trigger by the noise when the capacitor is fully charged. Our current prototype uses only a capacitor filtering of noise, but a low-pass filter, for example, can be considered to achieve better results. The so-called RC filter is a good example of such a low-pass filter, and it only needs an additional resistor between the capacitor and the signal output pin of the RF receiver. Our prototype uses a polyester film box capacitor, which is sized by empirical testing.

The rotating trigger detection system is feasible only for global shutter cameras. In the case of rolling shutter cameras, the pointer bends in the images because of different exposure synchronization within an image frame (see Figure 16). The use of such cameras would require a rolling shutter correction (e.g., Grundmann et al., 2012; Ito and Okatani, 2016). However, even with such a correction, it is expected that the accuracy is not equal to global shutter cameras. In our case, the main objective was to evaluate the triggering system, and therefore we did not focus on implementing a rolling shutter correction but instead used easier global shutter cameras.

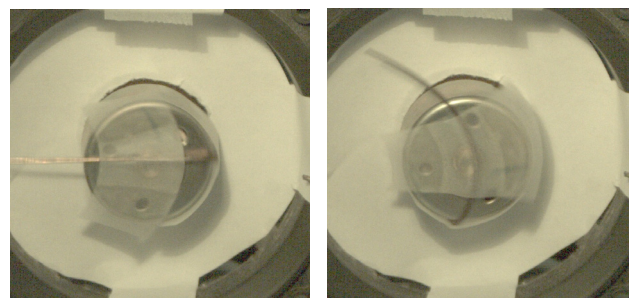


Figure 16. The behavior of a rolling shutter camera in the case of a rotating target. Because the view is scanned vertically with this camera, an almost horizontal moving pointer looks relatively straight (left image), whereas a moving vertical pointer bends strongly (right image).

The results include camera-based triggering errors, which can vary. Therefore, it is not obvious how much of the error is trigger-based and how much is camera-based. Minimum and maximum trigger time differences (see Table 1) should be interpreted cautiously. However, we believe it is safe to say that using a wireless triggering system with two different frequencies introduces additional trigger-based synchronization variation. This might be due to cross-talk between receivers of different frequencies. Interestingly, other wireless setups have slightly smaller mean trigger differences than wired systems. In addition, systematic errors can be corrected from the camera side by adding a short delay to the camera that triggers first, further improving the results.

Our trigger system can be extended to cover other commercial cameras and other electronic devices because the Arduino

offers a very flexible but straightforward way to implement various triggering scenarios and can be further extended to include other functions such as a serial data connection through radio transmitter/receiver pairs used to trigger auxiliary devices. Even if industrial cameras can be triggered directly from an Arduino data pin, for wired triggering of commercial cameras an additional transistor is needed, unfortunately. However, an Arduino data pin can be utilized to saturate the transistor, which in turn shorts the camera trigger pin to ground. Certain camera manufacturers also offer functions such as focus and zoom control, and some cameras need a certain type of wake-from-sleep signal in order to accept remote triggering at all. Arduino platforms can be programmed to cope with all such different situations. Our prototype uses a one-direction operation, but it could be extended to cope with bi-directional communication by adding an Arduino controller to the receiver side as well. One example of such operation would be to read the timestamp of the camera trigger and send it back to the Arduino controller to make sure the synchronization was correct in all cameras.

7. CONCLUSIONS

Arduino appeared to be a flexible and efficient environment to implement camera-triggering systems, and it is readily extendable by various add-on modules developed by people from all around the world. Our aim was to make Arduino-based wired and wireless triggering systems and to evaluate their synchronization accuracy. For evaluation purposes, a prototype of a rotating trigger detection system (named RoTriDeS) was developed.

The evaluation of triggering systems revealed that wireless triggering systems performed as well or even better than wired systems when only one frequency was applied for signal transmission. However, when two different frequencies were applied, a small trigger-based delay was detectable.

Overall, the triggering time differences were relatively small in all cases. The mean triggering differences varied between 7.92 and 9.42 μs , and the maximum difference of all systems was 29.31 μs . The maximum mean difference between systems was 1.417 μs when the mean time difference values of the different cases were compared. This research indicated that Arduino-based triggering systems are feasible and can be implemented wire-based or wirelessly.

8. ACKNOWLEDGEMENTS

This article was supported by the Academy of Finland, the Centre of Excellence in Laser Scanning Research, CoE-LaSR (No. 272195) and project “COMBAT” (293389). The Finnish Funding Agency for Innovation is acknowledged for support in project “VARPU” (7031/31/2016).

9. REFERENCES

Cronk, S., Fraser, C.S., Hanley, H.B., 2006. Automatic calibration of colour digital cameras. *Photogrammetric Record*, 21 (116), pp. 355-372.

Grundmann, M., Kwatra, V., D. Castro, D., 2012. Calibration-free rolling shutter removal. In: *Proc. ICCP*, 2012, 8 pages.

Ito, E., Okatani, T., 2016. Self-calibration-based Approach to Critical Motion Sequences of Rolling-shutter Structure from Motion. *CoRR*, abs/1611.05476, 9 pages.

Litos, G., Zabulis, X., Triantafyllidis, G., 2006. Synchronous Image Acquisition based on Network Synchronization. In: *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, 6 pages.

Luhmann T., 2014. Eccentricity in Images of Circular and Spherical Targets and its Impact to 3D Object Reconstruction. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. xl-5, pp. 363-370.

Shortis, M. R., Clarke, T. A., Robson, S., 1995. Practical Testing of the Precision and Accuracy of Target Image Centring Algorithms. *Videometrics IV, SPIE Vol. 2598*, pp. 65-76.

Thiel, D., Goulart, M. G., Botelho, S. S. C., Bem, R. A., 2013. Hardware and Software Infrastructure to Image Acquisition using Multiple Cameras. In: *Proceedings of SIBGRAPI 2013 (XXVI Conference on Graphics, Patterns, and Images)*, pp. 1-6.