Synchronous Image Acquisition based on Network Synchronization

Georgios Litos, Xenophon Zabulis and Georgios Triantafyllidis
Informatics and Telematics Institute
Thessaloniki, Greece
{gl,xenophon,gatrian}@iti.gr

Abstract

In this paper, a software-based system for the real-time synchronization of images captured by a low-cost camera framework is presented. It is most well suited for cases where special hardware cannot be utilized (e.g. remote or wireless applications) and when cost efficiency is critical. The proposed method utilizes messages to establish a consensus on the time of image acquisition and NTP synchronization of computer clocks. It also provides with an error signal, in case of failure of the synchronization. The evaluation of the proposed algorithm using a precise LED array system (1ms accuracy) proves the effectiveness of this method.

1. Introduction

The problem of camera synchronization was one of the first significant issues that have been raised with the production of the camera in the beginning of the 20th century.

Since 1923 and for the needs of the silent film production, F.H. Richardson [1], stated clearly the importance of synchronizing taking and camera speeds: “Any departure from perfect synchronization of the taking, or camera speed, and the speed of reproduction, or projection, must, and inevitably will cause the moving object to appear differently upon the screen than it appeared to and was photographed by the camera, hence under such conditions the spectator cannot and will, not see the moving object as the camera saw it.”

From this problem of synchronization for the silent film production, nowadays, we have moved to a similar issue in the area of the stereoscopic and multi-view imaging since the vast majority of modern computer vision techniques require the synchronized image acquisition up to the millisecond in real-time.

A powerful multi-PC computing platform is needed since the simultaneous (synchronized) acquisition and processing of multiple video sequences requires data transfer rates that are far above the specifications of a standard workstation since the computer buses as well as the camera connection buses (firewire, USB, etc) cannot serve the high data transfer needs of a multicamera application. A way to overcome this limitation is to use a distributed PC system, where each unit handles one or more cameras.

In this paper, a software-based system for the real-time synchronization of images captured by a low-cost camera framework is presented. It is highly recommended for cases where special hardware cannot be used. Thus, this system is applicable where a remote multicamera platform is used for the synchronized image acquisition. Examples of these applications include a multicamera system to monitor airplanes in the airport, a wireless camera network etc.

The system consists of two software parts: The camera servers and a client. The camera server runs on each computer that has one or more cameras attached to it. The client can run on any machine with a network connection. Capturing is controlled by the client. The proposed approach provides synchronization with an error-checking constraint. The evaluation of the proposed algorithm using a precise LED array system (1ms accuracy) proves the effectiveness of this method.

The remainder of the paper is organized as follows: In section 2 several approaches for the problem of the multi-camera synchronization are presented. Section 3 introduces the proposed method to synchronized image acquisition for a cluster of cameras. In section 4, experimental results are illustrated and the proposed method is evaluated. Finally, conclusions are drawn in section 5.

2. Background Work

Several approaches to camera synchronization can be found in the literature. They can be sorted under the categories of (a) special-purpose hardware, (b) “post-processing” synchronization algorithms, (c) Network synchronization, (d) software-based methods.

The most common one is to use special-purpose hardware. It usually consists of a microcomputer control unit which is dedicated to propagating external synchronization signals for triggering the cameras and achieving the cameras synchronization.

Point Grey Research Inc. manufactured the Sync Unit [2] which synchronizes the image acquisition of multiple cameras on different IEEE-1394 buses, either within the same computer or across multiple computers. Without a Sync Unit, there is no timing correlation between separate cameras on separate buses. The Objective Imaging OASIS-DC1 [3] is another board controller, providing synchronization for various digital cameras supporting trigger output signals. In [4], the authors introduce a single chip microcomputer control system for the synchronization operation of several rotating high-speed cameras. The proposed system consists of four parts: the microcomputer control unit (including the synchronization part and precise measurement part and the time delay part), the shutter control unit, the motor driving unit and the high voltage pulse generator unit. Several approaches for synchronization at the hardware level are proposed in [5] using either specialized cameras or external dedicated electric signal. All these hardware-based solutions properly and precisely synchronize the cameras but they are also potentially costly, technically complex and not very flexible (not applicable for many applications).

An alternative approach for the camera synchronization is based on various “post-processing” algorithms which are applied on unsynchronized sequences and try to find the temporal offset between them. Thus, knowing the time-shifts between enough view pairs, the whole network can be synchronized. In [6], Sinha et al proposed an automatic approach to synchronize a network of uncalibrated and unsynchronized video cameras by computing the epipolar geometry from dynamic silhouettes and finding the temporal offset between them. This is then used to compute the fundamental matrices and the temporal offsets between many view-pairs in the network.

A new method to synchronize video recording the same scene from different viewpoints is presented in [7]. This method relies on correlating space-time interest point distribution in time between videos. Space-time interest points represent events in video that have high variation in both space and time. Authors show that by detecting, selecting spacetime interest points and correlating their distribution, videos from different viewpoints can be automatically synchronized. Another similar approach is presented in [8] where the proposed approach models the asynchronism of the videos within the photogrammetric analysis.

These post-processing methods are quite effective in automatically recovering the frame temporal offset between image sequences and thus enabling the “post-synchronization” of the cameras. However, these algorithms cannot be applied real-time and assume a constant temporal offset. Furthermore such approaches are sensitive to occlusion, since they rely on the tracking of image features; which can be hidden by another imaged surface at arbitrary time frames.

Interconnecting the computers that the cameras are attached to through a local area network (LAN) facilitates their software synchronization, using network protocols, such as NTP (network time protocol) [9].This is a simple procedure, but experiments show that is not more efficient in camera synchronization compared to the methods using special-purpose hardware.

Recently, some software-based methods have been proposed for camera synchronization, which utilize cost-efficient and off-the-shelf cameras that do not allow synchronization through external triggering.

In [10], the proposed system consists of the camera-computers and the triggering-computer. This triggering computer can launch simultaneously all the cameras. Then, the cameras will immediately start capturing an image. Since there is not any proposal to handle the situation in case of failure of the synchronization, the method is dependent on the quality of the Ethernet connection, the operating system’s latency of response to the received triggering signal, as well as the camera drivers and hardware.

In [11], a more sophisticated approach to software-based camera synchronization is introduced. This method uses the server-client architecture such as the previous method with a simple error-checking technique. The problem of the discrepancy in synchronization is solved by calculating the time for sending data over the network by sending some test data and calculates the network latency. Then this latency is added to the reference time and sent to all the computers which they accept this value as their time value thereby synchronizing their clocks with the true reference time. This technique improves the camera synchronization, but assumes a constant latency which is added to the reference time. This assumption does not always hold, since the network, software and hardware lag do not always sum to a constant value along time and, thus, the results for the camera synchronization are inaccurate.

A similar system is presented in [12] where the authors proposed an easy to set up, software based system to record synchronized multi-video data. This system sends only the start pulse of the recording and then leaving it up to each computer to individually keep the pace. Specifically, they proposed to synchronize the clocks of the computers by using a time synchronisation daemon, and then a future start-time is sent instead of a direct triggering message. However, no results are reported for the efficiency of this synchronization.

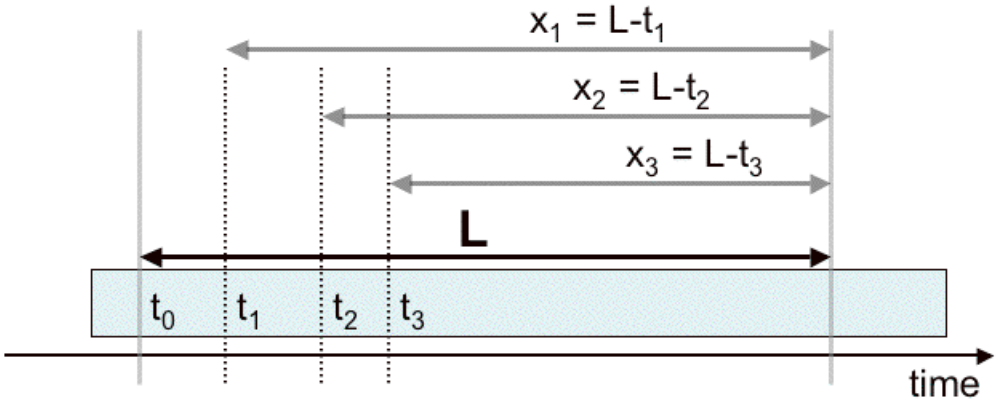


Figure 0 Illustration of the events occurring within the proposed approach to synchronous image acquisition.

In this work, a software-based client-server method is proposed for the efficient and real-time synchronization of a multi-camera network, without any need for special-purpose hardware or post-processing algorithm. It utilizes messages to establish a consensus on the time of image acquisition and NTP synchronization of computer clocks to attempt that this time for each computer corresponds to the same instant for all computers. It also provides with an error signal, in case of failure of the synchronization.

3. Theory

In this section, an approach to synchronized image acquisition for a cluster of cameras is introduced. This approach concerns cameras that are mounted on different computers which are, in turn, connected through a LAN. The method utilizes messages to perform the synchronization and provides also with an error signal, in case of failure of the synchronization. The method assumes that the computers are already synchronized by virtue of a NTP [9].

The proposed synchronization requires an estimate *L* of how much time is required for a message broadcasted by a client to reach all servers. Since the messages will contain a very small amount of data the *L* is assigned with an estimate of the LANs latency. Synchronized image acquisition is then achieved as follows.

Upon triggering, at time t_0, the client broadcasts a message to all servers. This message contains a time value is $t_a = t_0+L$, which corresponds to the time instance that the acquisition is planned to occur. The servers receive this message at different time instances. Upon reception of the message each server records the time into value t_i. Immediately after, they enter a busy waiting mode. The amount of time that each server will wait is: $x_i = t_i - t_0 - L$. Therefore, if servers are in synchronization, values t_i accurately represent the time differences in the reception of messages. Thus, values x_i will correspond to the same time instance and should be equal to t_a (see figure 1).

The proposed approach provides a synchronization error-checking constraint. If $x_i < 0$, then time interval L was too short for the LAN’s latency and, thus, is required to be increased. A software event is then generated signaling the failure of the synchronization process. This event can facilitate the automatic adjustment of *L*’s value, in order for the system to modulate the temporal frequency of image acquisition with respect to LAN latency, traffic, and bandwidth.

4. Experimental results

For the evaluation of the proposed method a PCB board is used which is based on DSP (MicroChip) embedded microprocessor with LEDs as an accurate chronometer providing 1 ms accuracy (see Figure 2). There are 4 rows of 10 LEDs that correspond to the following tempopral granuilarities: 1000ms, 100ms, 10ms, and 1ms.

The proposed method of evaluation using the PCB board provides better accuracy compared to the method of using a chronometer in a CRT screen. A CRT screen of 100 Hz provides approximately 10 ms accuracy.

Two identical workstations were used considered high-end at the time of the experiments, running at 3.2GHz with Firewire-800 PCI boards. Windows 2000 operating system was used due to incompatibilities with later versions with the firewire implementation. Although this is not a real-time OS, since our application runs with high-priority, we achieve high performance and real-time response times. The time-waiting algorithm is reliable because it is optimized for accuracy. Several time-counting methods were tested and the best one for the specific CPUs used in the experiment was chosen. The acquisition devices used were Unibrain’s Fire-i cameras connected on an IEEE1394 standard cable.

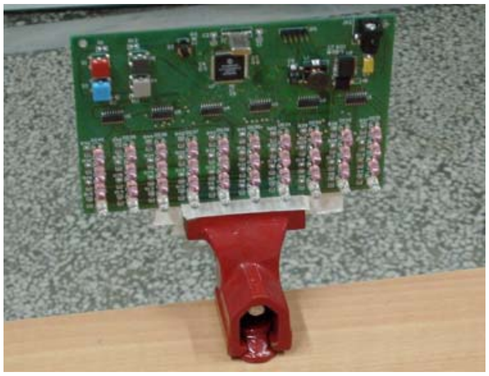


Figure 2 A PCB board based on DSP (MicroChip) embedded microprocessor with LEDs

Figure 3 shows a multi-camera system with four cameras targeting the PCB board. The experiment took place in a dark room. The shutter time of capture devices was reduced to 437μs to increase the light level and react better in sudden scene changes (of LEDs).

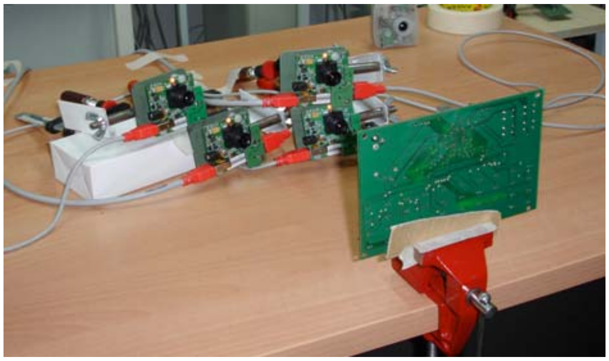


Figure 3 A multi-camera system with four cameras

The multi capture picture application framework consists of two parts (see Figure 4):

- The capture device station which powers the cameras, accept requests from control station and can be used as image storage system (see Figure 5a)
- The control station which controls the capture device stations, requests and gathers captured images from all stations (see figure 5b)

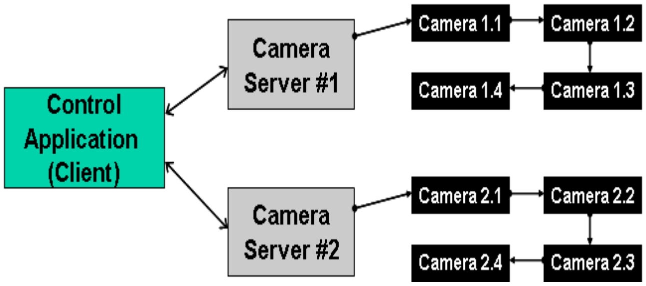
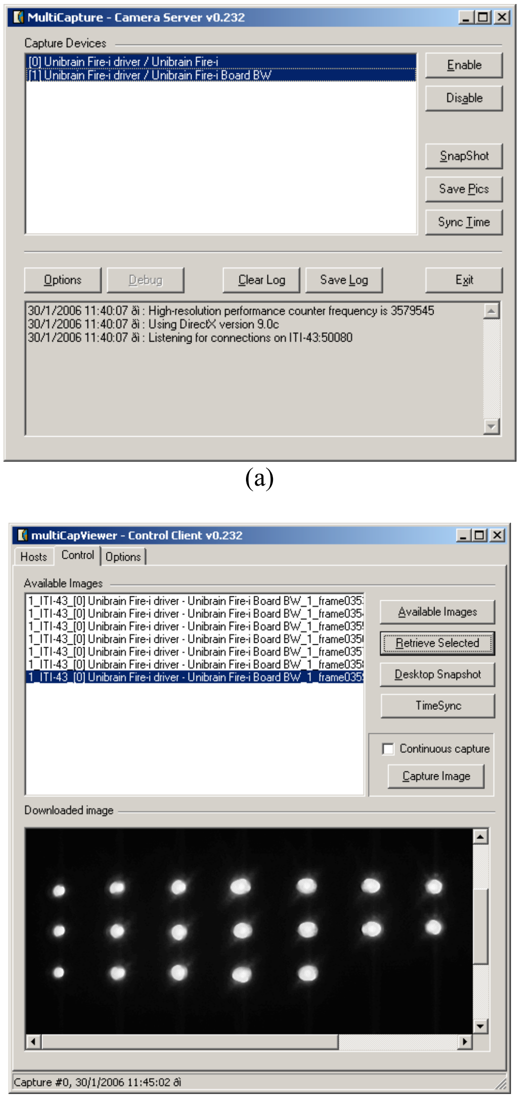


Figure 4 Multi camera capture architecture



MultiCapture - Camera Server v0.232

Capture Devices

[0] Unibrain Fire-i driver / Unibrain Fire-i
[1] Unibrain Fire-i driver / Unibrain Fire-i Board BW

```
30/1/2006 11:40:07 ăi : High-resolution performance counter frequency is 3579545
30/1/2006 11:40:07 ăi : Using DirectX version 9.0c
30/1/2006 11:40:07 ăi : Listening for connections on ITI-43:50080
```

(a)

multiCapViewer - Control Client v0.232

Available Images

1_ITI-43_[0] Unibrain Fire-i driver - Unibrain Fire-i Board BW_1_frame035
1_ITI-43_[0] Unibrain Fire-i driver - Unibrain Fire-i Board BW_1_frame035
1_ITI-43_[0] Unibrain Fire-i driver - Unibrain Fire-i Board BW_1_frame035
1_ITI-43_[0] Unibrain Fire-i driver - Unibrain Fire-i Board BW_1_frame035
1_ITI-43_[0] Unibrain Fire-i driver - Unibrain Fire-i Board BW_1_frame035
1_ITI-43_[0] Unibrain Fire-i driver - Unibrain Fire-i Board BW_1_frame035
1_ITI-43_[0] Unibrain Fire-i driver - Unibrain Fire-i Board BW_1_frame035

□ Continuous capture

Downloaded image

Capture #0, 30/1/2006 11:45:02 ăi

(b)

Figure 5 a) Camera Server, b) Control Client

Temporal synchronization of computers was retained by virtue of an SNTP server. The camera stations are synchronized at the start of each experiment. The network setup is based on a local 100mbit network (LAN) isolated from external networks. All cameras are identical so that we can assume that the devices have the same response time because it's a value that is cannot me measured. Capturing occurred at 30fps in the maximum resolution (640x480 pixels). The time interval (L) was 1500ms in our experiments (to avoid synchronization errors when taking large number of images) although it can be lowered to 500ms without problems. The proposed algorithm rejects the frames not taken at the correct intervals and reports the error to the user.

The accuracy we achieve by synchronizing each time two workstations using SNTP is between 0.000001ms and 0.000015ms (0.000005ms in average). The number is better than expected because in our case one workstation is acting as the SNTP server, therefore its time doesn't change and only one workstation has to be synchronized.

The goal of our experiments is to acquire images that can be accepted as captured at the same time frame in all camera stations. To test the efficiency, we measure the 'capture delay' between the camera systems. All cameras take pictures from the same target, our reference timer which is the LED-PCB board. The accuracy of the hardware counter is 1 ms. An application was developed to read pictures taken from the experiments and to extract the LED value using an optical recognition method. We count the difference in time (ms) between each acquisition system at the same capture command. The results give the total true delay including all the delays of network latency, of the synchronization using the NTP, of the cameras hardware and of the operating system reaction. Table 1 and Figures 6 and 7 summarize the experimental results.

Experiment	**Mean Value or Standard Deviation t**	**Minimum Latency**	**Maximum Latency**
Total 2 cameras 1 camera/PC	11.95 ms	0 ms	35 ms
Total 4 cameras 2 cameras/PC	9.988 ms	0 ms	92 ms

Table 1

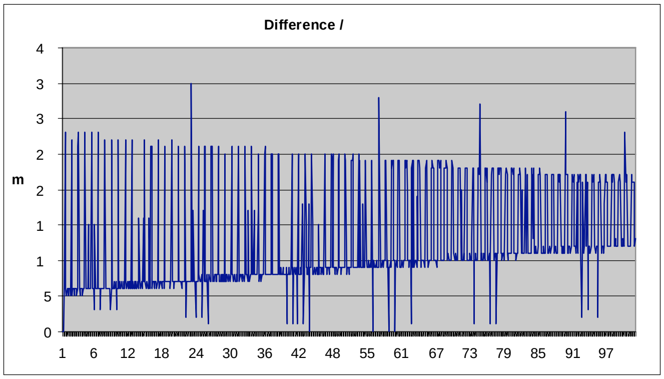


Figure 6 Difference in time (ms) between each acquisition system for the case of two cameras setup

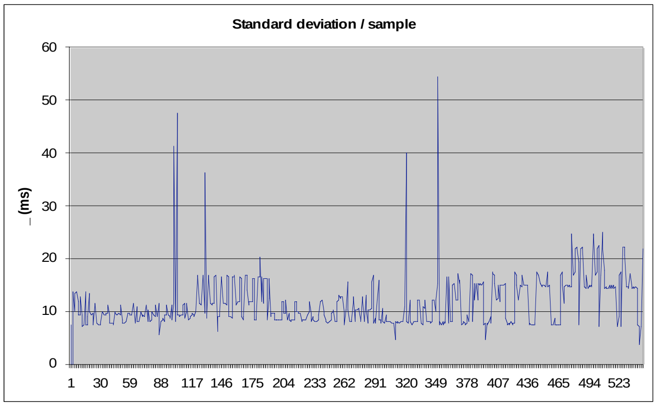


Figure 7 Standard deviation in time (ms) between each acquisition system for the case of four cameras setup

5. Conclusions

Although providing the most precise synchronization, external triggering requires special cameras that can receive and recognize such input as well as additional hardware and wiring. This increases the camera's cost and casts hardware-synchronization weak in term of cost-efficiency, since the cost of such components could be unreachable in several types of applications.

Moreover, in several types of applications independence from physical links (cable) is required due to the large distance between the cameras (e.g. ground traffic control, etc). This independence can also facilitate the synchronization of cameras mounted on wireless stations, suitable for mobile applications.

Therefore, software-based camera synchronization constitutes a useful alternative for a wide range of applications.

In this work a software-based client-server method is introduced for the real-time synchronization of a multi-camera network, which does not require any special-purpose hardware or post-processing algorithm. The advantages of this approach are: a) it is software-based, b) it can achieve real-time camera synchronization, c) it is cost-efficient, d) it is suitable for mobile applications and e) it is easy to set up since there is no need for special-purposed hardware.

The proposed approach is based on the precision with which the utilized NTP synchronization is achieved. Future work will temporally interpolate video-streams to predict the perfectly synchronous images that would have been acquired and also predict network latency to improve temporal synchronization.

Acknowledgement

The authors are grateful for support through the 3DTV Network of Excellence, 6th Framework, IST Programme.

References

[1] F.H. Richardson, "Importance of Synchronizing Taking and Camera Speeds," Transactions of S.M.P.E., No. 17, 1924, pages 117-123, October 1-4, 1923.

[2] Sync Unit, Point Grey Research Inc., http://www.ptgrey.com/products/sync/index.html

[3] OASIS-DC1 Digital Camera Trigger Interface, http://www.objectiveimaging.com/OASIS_DC1.htm

[4] Ningwen Liu, Yunfeng Wu, Xianxiang Tan, Guoji Lai, “Control system for several rotating mirror camera synchronization operation”, 22nd International Congress on High-Speed Photography and Photonics; Dennis L. Paisley, ALan M. Frank; Eds., Proc. SPIE Vol. 2869, p. 695-699, May 1997.

[5] Bertrand Holveck, Hervé Mathieu, “Infrastructure of the GrImage experimental platform: the video acquisition part”, Technical Report RT-0301, INRIA, Number RT-0301, Nov 2004.

[6] Sinha, S. N., Pollefeys, M.: Synchronization and Calibration of Camera Networks from Silhouettes. International Conference on Pattern Recognition ICPR, Vol. I, p. 116-119, 2004.

[7] Yan, J., Pollefeys, M., “Video Synchronization via Space-Time Interest Point Distribution”, Advanced Concepts for Intelligent Vision Systems ACIVS, 2004.

[8] Raguse, K., Heipke, C., “Photogrammetric analysis of asynchronously acquired image sequences”, In: Grün, A. ; Kahmen, H. (Hrsg.): Optical 3-D Measurement Techniques VII,. Band II, p. 71-80, 2005.

[9] Mills, D.L. Internet time synchronization: the Network Time Protocol. IEEE Trans. Communications COM-39, vol. 10, p. 1482-1493, Oct 1991.

[10] Tomas Svoboda, Hanspeter Hug, and Luc Van Gool, “ViRoom -- low cost synchronized multicamera system and its self-calibration”, In Pattern Recognition, 24th DAGM Symposium, pages 515-522. Springer, September 2002.

[11] Piyush Kumar Rai, Kamal Tiwari, Prithwijit Guha, Amitabha Mukerjee, “A Cost-effective Multiple Camera Vision System using FireWire Cameras and Software Synchronization”, 10th International Conference on High Performance Computing (HiPC 2003), Hyderabad, India, Dec. 17-20, 2003.

[12] Lukas Ahrenberg, Ivo Ihrke and Marcus Magnor, “A Mobile System for Multi-Video Recording”, IEE 1st European Conference on Visual Media Production (CVMP), London, UK, March 2004