Cost-Effective Solution to Synchronised Audio-Visual Data Capture using Multiple Sensors

J.F.Lichtenauer*[,a], J.Shen[a], M.F.Valstar[a], M.Pantic**[,a,b]

[a] *Department of Computing, Imperial College London, SW7 2AZ, United Kingdom*
[b] *Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands*

Abstract

Applications such as surveillance and human behaviour analysis require high-bandwidth recording from multiple cameras, as well as from other sensors. In turn, sensor fusion has increased the required accuracy of synchronisation between sensors. Using commercial off-the-shelf components may compromise quality and accuracy, because it is difficult to handle the combined data rate from multiple sensors, the offset and rate discrepancies between independent hardware clocks, the absence of trigger inputs or -outputs in the hardware, as well as the different methods for timestamping the recorded data. To achieve accurate synchronisation, we centralise the synchronisation task by recording all trigger- or timestamp signals with a multi-channel audio interface. For sensors that don't have an external trigger signal, we let the computer that captures the sensor data periodically generate timestamp signals from its serial port output. These signals can also be used as a common time base to synchronise multiple asynchronous audio interfaces. Furthermore, we show that a consumer PC can currently capture 8-bit video data with 1024x1024 spatial- and 59.1Hz temporal resolution, from at least 14 cameras, together with 8 channels of 24-bit audio at 96kHz. We thus improve the quality/cost ratio of multi-sensor systems data capture systems.

Key words: Video recording, Audio recording, Multisensor systems,

*Principal corresponding author

**Corresponding author

Email addresses: `j.lichtenauer@imperial.ac.uk` (J.F.Lichtenauer), `jie.shen07@imperial.ac.uk` (J.Shen), `Michel.Valstar@imperial.ac.uk` (M.F.Valstar), `m.pantic@imperial.ac.uk` (M.Pantic)

1. Introduction

In the past two decades, the use of CCTV (Closed Circuit Television) and other visual surveillance technologies has grown to unprecedented levels. Besides security applications, multi-sensorial surveillance technology has also become an indispensable building block of various systems aimed at detection, tracking, and analysis of human behaviour with a wide range of applications including proactive human-computer interfaces, personal wellbeing and independent living technologies, personalised assistance, etc. Furthermore, sensor fusion - combining video analysis with the analysis of audio, as well as other sensor modalities - is becoming an increasingly active area of research [1]. It is also considered a prerequisite to increase the accuracy and robustness of automatic human behaviour analysis [2]. Although humans tolerate an audio lag of up to 200ms or a video lag of up to 45ms [3], multimodal data fusion algorithms may benefit from higher synchronisation accuracy. For example, in [4], correction of a 40ms time difference, between the audio and video streams recorded by a single camcorder, resulted in a significant increase in performance of speaker identification based on Audio-Visual (A/V) data fusion. Lienhart et al. [5] demonstrated that microsecond accuracy between audio channels helps to increase signal separation gain in distributed blind signal separation.

With the ever-increasing need for multi-sensorial surveillance systems, the commercial sector started offering multi-channel frame grabbers and Digital Video Recorders (DVR) that encode video (possibly combined with audio) in real-time (e.g. see http://www.dvrsystems.net). Although these systems can be the most suitable solutions for current surveillance applications, they may not allow the flexibility, quality, accuracy or number of sensors required for technological advancements in automatic human behaviour analysis. The spatial and temporal resolutions, as well as the supported camera types of real-time video encoders are often fixed or limited to a small set of choices, dictated by established video standards. The accuracy of synchronisation between audio and video is mostly based on human perceptual acceptability, and could be inadequate for sensor fusion. Even if A/V synchronisation accuracy is maximised, an error below the time duration between subsequent video frame captures can only be achieved when it is exactly known how the recorded video frames correspond to the audio samples. Furthermore, commercial solutions are often closed systems that do not allow the accuracy of synchronisation that can be achieved with direct connections between the sensors. Some systems provide functionality of time-stamping the sensor data with GPS or IRIG-B modules. Such modules can provide microsecond synchronisation accuracy between remote systems. However, the applicability of such a solution depends on sensor hard- and software, as well as on the environment (GPS receivers need an unblocked view to the GPS satellites orbiting the Earth). Also, actual accuracy can never exceed the uncertainty of the time lag in the I/O process that precedes time-stamping of sensor data. For PC systems, this can be in the order of milliseconds [5].

A few companies aim at custom solutions for applications with requirements that cannot be met with what is currently offered by commercial surveillance hardware. For example, Boulder Imaging (www.boulderimaging.com) builds custom solutions for any application, and Cepoint Networks offers professional video equipment such as the Studio 9000$^{\mathrm{TM}}$ DVR (http://www.cepoint.com), which can record up to 4 video streams per module, as well as external trigger events, with an option to timestamp with IRIG-B. It also has the option of connecting an audio interface through a Serial Digital Interface (SDI) input. However, it is not clear from the specifications if the timestamping of audio and video can be done without being affected by the latency between the sensors and the main device. Furthermore, when more than 4 video streams have to be recorded, a single Studio 9000 will still not suffice. The problem of the high cost of custom solutions and specialised professional hardware is that it keeps accurately synchronised multi-sensor data capture out of reach for most computer vision and pattern recognition researchers. This is an important bottleneck for research on multi-camera and multi-modal human behaviour analysis. To overcome this, we propose solutions and present findings regarding the two most important difficulties in using low-cost Commercial Off-The-Shelf (COTS) components: reaching the required bandwidth for data capture and achieving accurate multi-sensor synchronisation.

Fortunately, recent developments in computer hardware technology have significantly increased the data bandwidths of commercial PC components, allowing for more audio-visual sensors to be connected to a single PC. Our low-cost PC configuration facilitates simultaneous, synchronous recordings of audio-visual data from 12 cameras having 780x580 pixels spatial resolution and 61.7fps temporal resolution, together with eight 24-bit 96kHz audio channels. By using six internal 1.5TB Hard Disk Drives (HDD), 7.6 hours of continuous recordings can be made. With a different motherboard and an extra HDD controller card to increase the amount of HDDs to 14, we show that 1 PC is capable of continuously recording from 14 gigabit ethernet cameras with 1024x1024 pixels spatial resolution and 59.1 fps, for up to 6.7 hours. In Table 1 we show the maximum number of cameras that can be used in the different configurations that we tested. A higher number of cameras per PC means a reduction of cost, complexity as well as space requirements for visual data capture.

Synchronisation between COTS sensors is hindered by the offset and rate discrepancies between independent hardware clocks, the absence of trigger inputs or -outputs in the hardware, as well as different methods of timestamping of the recorded data. To accurately derive synchronisation between the independent timings of different sensors, possibly running on multiple computer systems, we centralise the synchronisation task in a multi-channel audio interface. For sensors with an external trigger, we record the trigger signals directly into a separate audio track, parallel to tracks with recorded sound. For sensors that don't have an external trigger signal, we let the computer that captures the sensor data periodically generate timestamp signals from its serial port output. These signals can be recorded in a parallel audio channel as well, and can even be used as a common time base to synchronise multiple asynchronous audio interfaces.

Using low-cost COTS components, our approach still achieves a high synchronisation accuracy, allowing a better trade-off between quality and cost. Furthermore, because synchronisation is achieved at the hardware level, separate software can be used for the data capture from each sensor. This allows the use of COTS software, or even freeware, maximising the flexibility with a minimal development time and cost.

The remainder of this article consists of five parts. We begin with describing related multi-camera capture solutions that have been proposed before, in section 2. In section 3, we describe the choices that need to be made for components in a multi-sensor data capture system. Experimental results of the system throughput are described in section 4. In section 5 we describe procedures for synchronisation between sensors with and without an external trigger. Section 6 contains our conclusions about the achieved knowledge and improvements.

Table 1: Camera support of a single consumer PC

Spatial Resolution	**Temporal Resolution**	**Rate per Camera**	**Max. Nr. of Cameras**
780x580 pixels	61.7fps	26.6MB/s	14
780x580 pixels	49.9fps	21.5MB/s	16
780x580 pixels	40.1fps	17.3MB/s	18
With controller card for 8 additional HDDs			
1024x1024 pixels	59.1fps	59.1MB/s	14

2. Related Work

Because of the shortcomings and high costs of commercially available video capture systems, many researchers have already sought custom solutions that meet their own requirements.

Zitnick et al. [6] used two specially built concentrator units to capture video from eight cameras of 1024x768 pixels spatial resolution at 15fps.

Wilburn et al. [7] built an array of 100 cameras, using 4 PCs and custom-built low-cost cameras of 640x480 pixels spatial resolution at 30fps, connected through trees of interlinked programmable processing boards with on-board MPEG2 compression. They used a tree of trigger connections between the processing boards (that each control one camera) to synchronise the cameras with a difference of 200 nanoseconds between subsequent levels of the tree. For a tree of 100 cameras, this should result in a frame time difference of 1.2μs, between the root and the leaf nodes.

More recently, a modular array of 24 cameras (1280x1024 pixels at 27fps) was built by Tan et al. [8]. Each camera was placed in a separate special-built hardware unit that had its own storage disk, using on-line video compression to reduce the data. The synchronisation between camera units was done using a tree of trigger- and clock signal connections. The delay between the tree nodes was not reported. Recorded data was transmitted off-line to a central PC via a TCP/IP network.

Svoboda, et al. [9] proposed a solution for synchronous multi-camera capture involving standard PCs. They developed a software framework that manages the whole PC network. Each PC could handle up to three cameras of 640x480 pixels spatial resolution at 30fps, although their software was limited to handling a temporal resolution of 10fps. Camera synchronisation was done by software triggers, simultaneously sent to all cameras through the ethernet network. This solution could reduce costs by allowing the use of low-cost cameras that do not have an external trigger input. However, the cost of multiple PCs remains. Furthermore, a software synchronisation method has a much lower accuracy than an external trigger network.

A similar system was presented in [10], which could handle 4 cameras of 640x480 pixels spatial resolution at 30fps per PC. The synchronisation accuracy between cameras was reported to be within 15 milliseconds.

Hutchinson et al. [11] used a high-end server PC with three Peripheral Component Interconnect (PCI) buses that provided the necessary bandwidth for 4 FireWire cards and a PCI eXtended (PCI-X) Small Computer System Interface (SCSI) Hard Disk Drive (HDD) controller card connecting 4 HDDs. This system allowed them to capture video input from 4 cameras of 658x494 pixels spatial resolution at 80fps.

Fujii et al. [12] have developed a large-scale multi-sensorial setup capable of capturing from 100 cameras of 1392x1040 pixels spatial resolution at 29.4fps, as well as from 200 microphones at 96kHz. Each unit that captures from 1 camera (connected by a Camera Link interface), and 2 microphones, consists of a PC with custom built hardware. During recording, all data is stored to internal HDDs, to be transported off-line via ethernet. A central host computer manages the settings of all capture units as well as the synchronous control unit that generates the video- and analog trigger signals from the same clock. By using a single, centralised trigger source for all measurements, the synchronisation error between sensors is kept below 1μs. Disadvantages of this system are the high cost and volume of the equipment, as well as the required custom built hardware.

Table 2 summarises the multi-camera capture solutions that we have described above. From this, it immediately becomes clear that audio has been a neglected factor in previous multi-sensor data capture solutions. With custom hardware, only Fujii et al. achieve accurate A/V synchronisation. The only low-cost solution that has a standard support for audio is a commercial surveillance DVR system. Unfortunately, having a microsecond synchronisation accuracy is not a key issue in surveillance applications, since the primary purpose of the systems is to facilitate playback to a human observer. However, having such a punctilious synchronisation accuracy is necessary for achieving (automatic) analysis of human behaviour.

To the best of our knowledge, the multi-sensor data capture solution proposed here is the first complete multi-sensor data capture solution that is based on commercial hardware, while achieving accurate synchronisation

Table 2: Overview of multi-sensor audio-visual data capture solutions. A ‘unit’ is a system in which sensor data is collected in real-time. For most cases, this is a PC. However, for Zitnick et al. [6] it was a ‘concentrator unit’. ‘camera#/unit’ indicates the maximum number of cameras that can be connected to a unit, ‘audio ch#/unit’ indicates the maximum number of audio channels per unit, ‘sync unit#’ shows the maximum number of units that can be synchronised, ‘unit sync’ the type or accuracy (if known) of synchronisation between units, ‘camera sync’ the the type or accuracy of synchronisation between cameras and ‘A/V sync’ the accuracy of synchronisation between audio and video.

description	**camera#/unit at 640x480 30fps**	**audio ch#/unit**	**sync unit#**	**unit sync**	**camera sync**	**A/V sync**
Our solution	14× 1024x1024p at 59.1fps	≤7	unlimited	<20μs	∼30μs	∼25μs
Studio 9000 DVR	4	optional via SDI input	unlimited with IRIG-B	optional IRIG-B	depends on the cameras	not specified
typical surveillance DVR	16	16	1	n.a.	depends on the cameras	not specified
Zitnick et al. [6]	4× 1024x768p at 15fps	n.a.	≥2 (not specified)	by firewire	not specified	n.a.
Wilburn et al. [7]	30	n.a.	unlimited	hardware trigger	1.2μs with 100 cameras	n.a.
Tan et al. [8]	1× 1280x1024p at 27fps	n.a.	unlimited	hardware trigger	hardware trigger	n.a.
Svoboda et al. [9]	3 at 10fps	n.a.	unlimited	network trigger	software trigger	n.a.
Cao et al. [10]	4	n.a.	unlimited	15 ms w. 16 units	software trigger	n.a.
Hutchinson et al. [11]	4× 658x494p at 80fps	n.a.	1	n.a.	software trigger	n.a.
Fujii et al. [12]	1× 1392x1040p at 29.4fps	4	unlimited	<1μs with 100 units	<1μs with 100 cameras	<1μs with 100 units

between audio and video, as well as with other sensors and computer systems.

3. System Design

This section describes the components of our system and explains the motivations behind the most important design choices that we had to make. Although the focus of this paper is on achieving sufficient capture bandwidth and synchronisation accuracy with COTS components, the design aspects common to all audio-visual data capture systems cannot be overlooked to obtain a solution that meets the requirements of a human behaviour analysis application.

The relevant components of our system setup are summarised in Table 3. We will start with the visual capture aspects, including resolution, shutter, colour, lens- and sensor size, synchronisation, followed by the interface between cameras and the PC, the illumination and possible post-processing steps. Subsequently, we describe the audio sensors, the computer hardware for the data storage, and the utilised motherboard. Finally, we describe the software we used for audio and video capture.

Table 3: Components of the capture system for 8 FireWire cameras with a resolution of 780x580pixels and 61.7fps

Sensor Component	**Description**
7 monochrome video cameras	AVT Stingray F-046B, 780x580 pixels resolution, max. 61.7fps
colour video camera	AVT Stingray F-046C, 780x580 pix. Bayer pattern, max. 61fps
2 camera interface cards	Dual-bus IEEE 1394b PCI-E×1, Point Grey
room microphone	AKG C 1000 S MkIII
head-worn microphone	AKG HC 577 L
external audio interface	MOTU 8-pre Firewire 8-channel, 24-bit, 96kHz
Eye tracker	Tobii X120
Computer Component	**Description**
6 Capture disks	Seagate Barracuda 1.5TB SATA, 32MB Cache, 7,200rpm
System disk	PATA Seagate Barracuda 160GB 2MB Cache, 7,200rpm
Optical drive	PATA DVD RW
4GB Memory	2GB PC2-6400 DDR2 ECC KVR800D2E5/2G
Graphics card	Matrox Millenium G450 16MB PCI
Motherboard	Asus Maximus Formula, ATX, Intel X38 chipset
CPU	Intel Core 2 Duo 3.16GHz, 6MB Cache, 1333MHz FSB
ATX Case	Antec Three Hundred
PSU	Corsair Memory 620 Watt
Software Application	**Description**
MS Windows Server 2003	32-bit Operating System
Norpix Streampix 4	Multi-camera video recording
Audacity 1.3.5	Freeware multi-channel audio recording
AutoIt v3	Freeware for scripting of Graphical User Interface control
Tobii Studio version 1.5.10	Eye tracking & stimuli data suite
Tobii SDK	Eye tracker Software Development Kit

3.1. Spatial and temporal video resolution

The main properties to choose in a video camera are the spatial and temporal resolution. Selecting an appropriate spatial resolution involves essentially a trade-off between Signal-to-Noise Ratio (SNR) and the level of detail. Sensors with higher spatial resolution receive less light per photo sensor (due to smaller sensor sizes), and are generally less efficient (more vulnerable to imperfections and circuitry takes up relatively more size). These factors contribute to a lower SNR when a higher spatial resolution is used.

Furthermore, a higher spatial and/or temporal resolution is more costly. Not only that the high-resolution cameras are more expensive, but the required hardware capable of real-time data capture and recording of the high data rate is more expensive as well. Another issue that needs to be taken into consideration when a high temporal resolution is used, is the upper limit for the shutter time, which equals the time between video frames. Depending on the optimal exposure, high-speed video may require brighter illumination and more sensitive imaging sensors, in order to achieve a sufficient SNR.

For these reasons, it is crucial to choose no more than the minimum spatial and temporal resolution that provides sufficient detail for the target application. The analysis of temporal segments (onset, apex, offset) of highly-dynamic human gestures, such as sudden head and hand movements, demands a limited shutter time (to prevent motion blur) as well as sufficient temporal resolution (to capture at least a couple of frames of each gesture). Previous research findings in the field of dynamics of human behaviour reported that the fastest facial movements (blinks) last 1/4 seconds [13, 14], and that the fastest head and hand movements (finger movements) last 1/12 seconds [15]. Hence, in order to facilitate analysis of temporal segments of various gestures, we needed a camera with temporal resolution of at least 60 fps, facilitating capture of even the fastest gesture in at least 5 frames, with each temporal segment of the gesture captured in 1-2 frames. Figure 3 shows a fast head turn captured at 60fps.

3.2. Shutter

'Interlacing' or 'rolling shutter' sensors have an advantage in light efficiency and frame rate, but produce severe distortions of moving objects. This is shown in figure 1. For computer vision applications involving moving objects, such as human beings or parts of the human body, progressive scan global shutter sensors are the primary choice.

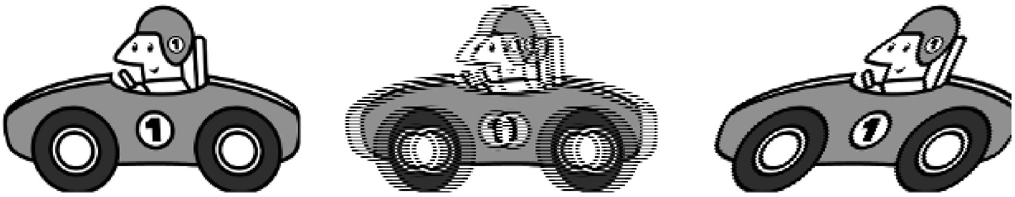

Figure 1: Example of how an image of a horizontal moving object looks like, when captured with a camera with (1) progressive scan with global shutter (left), (2) interlaced scan (middle) and (3) progressive scan with rolling shutter (right)

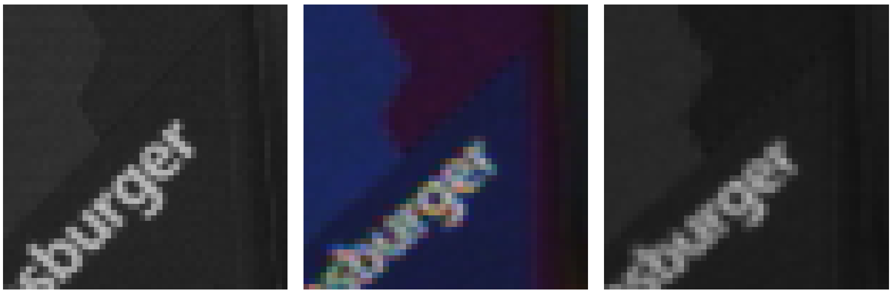

Figure 2: Comparison of the AVT Stingray F-046B monochrome camera with shutter 1/60s (left) to the AVT Stingray F-046C Bayer colour camera with shutter 1/20s (middle). The right image is obtained by converting the colour image to a grey image.

3.3. Colour vs. Monochrome

Most of the current colour cameras make use of a Bayer filter that passes either red, green or blue to each photo sensor on the imaging chip. Colour can be reconstructed by combining the values of adjacent pixels that represent different colours. In this way, a colour camera captures exactly the same amount of data as a monochrome camera. It is only after the demosaicing (which can be done off-line) that the amount of data is multiplied by three, to obtain a colour image. However, a Bayer filter has four main disadvantages.
1) The colour filter in front of the sensor blocks almost 2/3 of the incoming light. A monochrome camera needs only 1/3 of the shutter time for the same image intensity (resulting in 2/3 reduction of motion blur). Figure 2 shows how an image from a monochrome camera compares to a three times longer shutter time with a colour camera.
2) All pixels in the reconstructed image will depend on at least three different

locations in the RAW Bayer pattern, reducing sharpness. A grey image from a monochrome sensor is almost twice as sharp, compared to a Bayer reconstruction (see figure 2).
3) Colours are reconstructed incorrectly around edges.
4) 'Binning' of Bayer patterns is not possible. The binning functionality of a monochrome camera (if supported) divides the resolution of a camera by 2 and increases SNR by $\sqrt{2}$ in horizontal and/or vertical direction. This is useful to reduce data rate during the data capture process, when the full image resolution is not required.

Therefore, the choice between a colour or monochrome camera involves a trade-off between these disadvantages and the added value of colour information. Instead of a Bayer pattern, some cameras utilise a prism that separates the colours onto three separate image sensors. However, these cameras only work with special lenses, reducing design choices and increasing the costs significantly. Another technology that eliminates the disadvantages of a Bayer filter is the 'Foveon X3 sensor' [16]. This image sensor has three layers of photo sensors on top of each other, with colour filters in between. Currently available industrial video cameras with this specific sensor are the Hanvision HVDUO-5M, -10M and -14M.

3.4. Lens and sensor size

Other important properties of the camera to be selected are the focal length and aperture. While the former is chosen in relation to the desired Field Of View (FOV), the latter is chosen for the desired Depth of Field (DoF) and/or shutter time. Figure 3 shows the effects of the trade-off between shutter time and DoF, where the images in the top row have the sharpest moving foreground, while the images in the bottom row, taken with smaller aperture and longer shutter time, have the sharpest background. Besides these basic optical properties, many other factors have to be taken into account, too. A lens is made for a specific camera mount and specific (maximal) sensor size. Therefore, when selecting a camera, the available lenses must be considered as well. For instance: a CS-mount camera with a 1/3" sensor will accept a C-mount lens (with an adaptor ring) specified for a 1/2" sensor, but not the other way around. The main advantage of a larger sensor size is that it generally results in less distortion of wide-angle views. However, this also greatly depends on the quality of a lens. Larger sensors also tend to have a better SNR. However, in practice, SNR depends more on the production technology than on the sensor size.

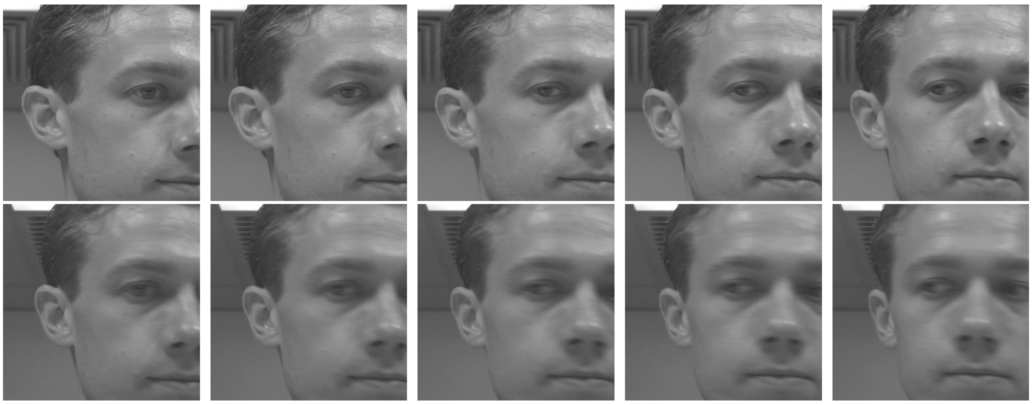

Figure 3: Example of two trade-offs between shutter time and aperture. The recorded action is a quick head turn as the result of a sudden change of attention. The images are cropped at 300x300pixels from a full resolution of 780x580pixels. The top row shows 5 subsequent images taken at 60fps, with a shutter time of 5ms. The bottom row shows images taken at the same moments, from a synchronised camera, with a shutter time of 15ms and a smaller aperture, to obtain the same image brightness. The result of the longer shutter time is an increased motion bur, while the smaller aperture results in a sharper background due to the increased DoF.

3.5. Camera Synchronisation

While software-triggering is a low-cost and simple solution for synchronising cameras, the architecture of general-purpose computer systems implies uncertainty in the arrival times of triggering messages, resulting in unsynchronised frame capture by different cameras. For some applications, this can still be sufficiently accurate. However, for stereo imaging and analysis of fast events by multi-sensor data fusion, hardware-triggering is demanded. Unfortunately, web-cams and camcorders generally do not support external triggering. This means that there isn't any choice but to use industrial cameras, which are generally in a higher price range. Note, however, that the limited image quality and capture control of web-cams makes them unsuitable for many applications anyway.

The AVT Stingray cameras, which we used in our multi-modal data capture system, provide a trigger input as well as output [17]. This facilitates building a relatively simple synchronisation network made out of up to 7 cameras (limited by the maximal output current of one camera), without any extra trigger- or amplification hardware. When the trigger output of the master camera is used as the input to the slave cameras, the resulting delay of the slave cameras is approximately 30μs. If more than 7 cameras must be synchronised, either a trigger amplifier/relay must be used, or the output of one of 6 slave cameras must be used as a trigger again, for 6 additional slave cameras. However, at each such step in the chain, another 30μs delay is added.

3.6. Camera Interface

The camera interface has an impact on the cost, the required bandwidth, the maximal number of cameras that can be connected to one PC, as well as on the CPU load [18]. The three main interfaces for machine-vision cameras are FireWire (400 or 800), 'GigE Vision' and 'Camera Link'.

FireWire (IEEE 1394) allows isochronous data transfer (74MB/s for IEEE 1394b with default channel settings). Isochronous data can be written directly to a Direct Memory Access (DMA) buffer by the FireWire bus controller, with a negligible CPU load. The maximum number of cameras that can be connected to one FireWire bus is typically limited to 4 or 8 (DMA channels), depending on the bus hardware. FireWire cameras can often be powered by the FireWire cable, which saves extra power supplies and cables for the cameras.

'GigE Vision' is an upcoming camera interface, based on Gigabit Ethernet (GbE), specifically standardised for machine vision. Depending on cameras, network configuration and packet loss, one GbE connection can support up to 100MB/s from multiple cameras. If many GigE cameras are connected to one PC, the CPU load can become significant. This can be reduced by using a special Network Interface Card/Chip (NIC) driver. A disadvantage of GbE, compared to FireWire, is that it is less trivial to combine multiple cameras on one channel. Collisions of packets from different cameras have to be prevented by setting packet transfer delays, or using expensive switches that can buffer the data and specify to GigE Vision requirements.

Camera Link (CL) is an interface that is specifically designed for high-bandwidth vision applications. CL is the only choice if a camera is required which generates a rate of data that exceeds the capacity of FireWire or GbE. Increases in bandwidth of FireWire and GbE, and the high cost of CL interface cards and cables, are making CL less attractive. With the upcoming of 10GbE and 100GbE networking, the bandwidth advantage of CL may be eliminated completely. Alternatively, some camera manufacturers are choosing to equip high-bandwidth cameras with multiple GbE connections (e.g.

the Prosilica GX-Series). Another reason to use CL is that it can provide a more deterministic image capture process [18], which can be important in time-critical applications where a system has to respond with low latency.

For our application that required cameras with a spatial resolution of 780x580 pixels and a temporal resolution of 60fps (25.9MB/s), we chose the IEEE 1394b interface. At the time of designing the setup, we were uncertain about the effective bandwidth and CPU load of the GigE Vision interface. Furthermore, FireWire was more common and allowed straightforward combining of two cameras on one port. For another application that required a resolution of 1024x1024 pixels and 60fps (60MB/s), we chose for the GigE Vision interface. With 60MB/s per camera, there would be no possibility to combine multiple cameras on one interface anyway. Furthermore, we needed to have at least 4 interface connections per expansion card, in order to support the required number of cameras in one PC. We found that GbE cards with 4 ethernet adapters were significantly cheaper than an IEEE 1394b card with 4 buses. Tests showed that the CPU load of the GigE Vision cameras didn't pose a problem in our setup.

3.7. Illumination

Illumination determines an object's appearance. The most important factors of illumination are spectrum, intensity, location, source size and stability.

3.7.1. Illumination Spectrum

If a colour camera is used, it is important that the light has significant power over the entire visible colour spectrum. If a monochrome camera is used, a monochrome colour source can improve image sharpness with low-cost lenses, by preventing chromatic abberation. Most monochrome cameras are sensitive to the Near Infra Red (NIR) wavelengths (between 700nm and 1000nm). Since the human eye is insensitive to these wavelengths, a higher illumination intensity can be used here (within safety limits), without compromising comfort. Furthermore, the human skin is more translucent to NIR light [19]. This has a smoothing effect on wrinkles, irregularities and skin impurities, which can be beneficial to some applications of computer vision.

3.7.2. Illumination Intensity

The intensity of light cast on the target object will determine the tradeoff between shutter time and noise. Short shutter times (to reduce motion blur) require more light. Light intensity may be either increased by a more powerful light source, or by focussing the illumination onto a smaller area (using focussing reflectors or lenses).

3.7.3. Illumination Source Location

For most machine-vision applications, the ideal location of the illumination source is at the position of the camera lens. There are many types of lens-mountable illuminators available for this. However, for human subjects, it can be very disturbing to have the light source in front of them. It will shine brightly into the subject's eyes, reducing the visibility of the environment, such as a computer screen. Placing the illumination more sideways can solve this problem. However, when a light source shines directly onto the glass of the camera lens, lens flare may be visible in the captured images. Especially in multi-camera data capture setups, these issues can cause design conflicts.

3.7.4. Illumination Source Size

Small (point) light sources cause the sharpest shadows, the most intense lens flare, and are the most disturbing (possibly even harmful) to the human eye. Therefore, in many situations, it is beneficial to increase the size of the light source. This can be either realised by a large diffuser between the light source and the subject, or by reflecting the light source via a large diffusing (white, dull) surface. Note that the size and shape of the light source will directly determine the size and shape of specular reflections in wet or glossy surfaces, such as the human eyes and mouth.

3.7.5. Illumination Constancy

For many computer-vision applications, as well as for data reduction in video compression, it is crucial to have constant illumination over subsequent images. However, the AC power frequency (usually around 50 or 60Hz) causes oscillation or ripple current in most electrically powered light sources. If the illumination cannot be stabilised, there are two alternative solutions to prevent 'flicker' in the captured video. The first is to use a shutter time that is equal to a multiple of the oscillation period. In case of a 100Hz period, the minimum shutter time is 10ms. In human behaviour analysis applications, this is not sufficiently short to prevent motion blur (e.g. by a fast moving hand). Another option is to synchronise the image capture with the illumination frequency. This requires special algorithms (e.g. [20]) or hardware (e.g. generating camera trigger pulses from the AC oscillation of the power source) and limits the video frame rate to the frequency of the illumination.

3.7.6. Illumination/Camera Trade-off

Experimenting with recordings of fast head and hand motions showed us that for a closeup video (where the inter-ocular distance was more than 100 pixels), the shutter time needs to be shorter than 1/200 seconds, in order to prevent significant motion blur. Obtaining high SNR with short shutter times requires bright illumination, a large lens aperture, or a sensitive sensor. Because illumination brightness is limited by safety and comfort of human beings, and the lens aperture is limited by the minimum required depth of field, video quality for human analysis depends highly on the sensor sensitivity. Therefore, it can be worthy investing in a high-quality camera, or sacrificing colour for the higher sensitivity of a monochrome camera.

3.8. Video post-processing

Depending on use of the image data, additional processing of recorded video may be required. Some camera models are able to perform a number of post-processing steps on-board already. We briefly describe the most common post-processing steps for computer vision applications:

Hot/cold pixel removal: Due to irregularities in sensor production, or the influence of radiation, some sensor locations have a defect that causes their pixel read-out values to be significantly higher (hot) or lower (cold) than the correct measurements. When these pixel locations are known by (frequent) testing of the camera, they can be ‘fixed’ either by compensating the value or by interpolation from the surrounding pixel measurements. For some camera models, irregularities from production are already compensated in the camera itself.

Vignetting correction: Angle-dependent properties of the lens and image sensor can cause a difference in brightness, depending on the location in the image. Usually, it is a gradual decrease of brightness from the centre to the edges of the image. Vignetting can be estimated and inverted.

Colour mapping: Mapping of pixel values can be necessary to compensate a non-linear intensity-response, to normalise intensity and contrast and/or, in the case of colour images, to achieve a correct white-balance or colour calibration.

Lens distortion correction: If accurate geometric measurements need to be performed on the images, the non-linear lens distortions can be estimated and inverted, to approximate the linear perspective distortion of a pinhole lens. For colour cameras, chromatic abberation can be reduced by using a different lens distortion correction for the red, green and blue channels.

Stereo rectification: If a large number of stereo disparity measurements have to be performed, it can be useful to convert the perspectives of a pair of cameras to simulate coplanar image planes with identical orientation. This causes all epipolar lines to be horizontal, thus aligned with pixel rows. Stereo rectification has to be combined with lens distortion correction.

Video compression: Video compression is required when the rate of raw video data becomes too large to be practical. Then, a trade-off between quality, speed and storage size needs to be made. Real-time video compression can be attractive to eliminate a time-consuming off-line compression step, or to capture to a storage device which is not capable of handling the rate of the raw video data. However, the efficiency of contemporary multi-pass compression methods with variable bit rate (e.g. as in H.264) is significantly higher than what can be achieved with real-time compression. Furthermore, hardware compression solutions are often limited to specific resolutions and frame-rates, and may be more costly than additional HDDs and HDD controller cards that can store the raw video data with a sufficiently high rate.

3.9. Microphones

Since many audio processing methods are vulnerable to noise, the microphone setup is an important factor for accurate multimodal data capture. Placing a microphone close to the subject's mouth will reduce background noise, but may occlude the subject's face or body. A head-mounted microphone with a small mouthpiece next to the cheek may provide a reasonable compromise for certain applications. When combined with a room (ambient) microphone, the person's voice recorded by a head-mounted microphone could be separated even better from background noise. Alternatively, a microphone array may be used to focus attention to a particular spatial location [21].

3.10. Storage

Currently, the Hard Disk Drive (HDD) is the most significant bottleneck of a conventional PC. Capturing to internal memory (RAM) is the best solution for short video fragments. However, many applications require significantly longer recordings than what can be stored in RAM. The fastest

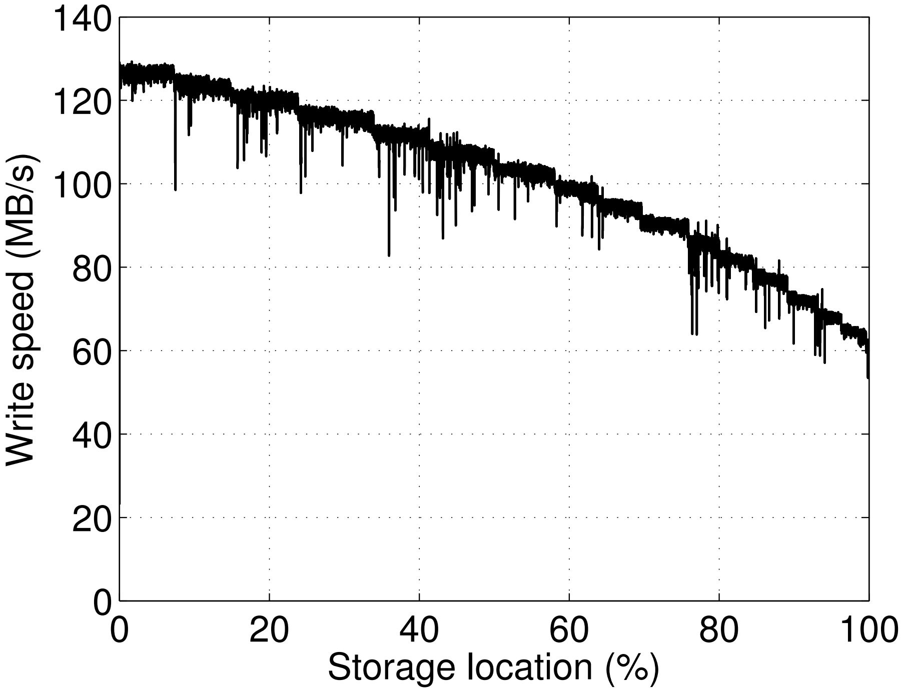

Figure 4: Sequential write transfer rate of 1.5TB Seagate Barracuda HDD as a function of disk location

consumer HDDs, with Serial Advanced Technology Attachment (SATA) interface, currently start with a data rate of over 100MB/s (at the outside of the platter) and gradually descent to a rate of around 60MB/s at the end of the disk. The decrease in Write Transfer Rate (WTR) of a 1.5TB Seagate Barracuda disk is shown in figure 4.

Most high-end consumer motherboards provide SATA connections for six disks, including hardware RAID support, which will allow a total capture rate of approximately 500MB/s (depending on how much of the disk space is used for capture). Video streams from multiple cameras can be either written to separate HDDs, or to a single RAID0 disk that consists of multiple physical member HDDs. A RAID0 disk has a size equal to the number of member disks (N) multiplied by the size of the smallest disk, and a WTR that comes close to $N\times$ the throughput of the slowest disk.

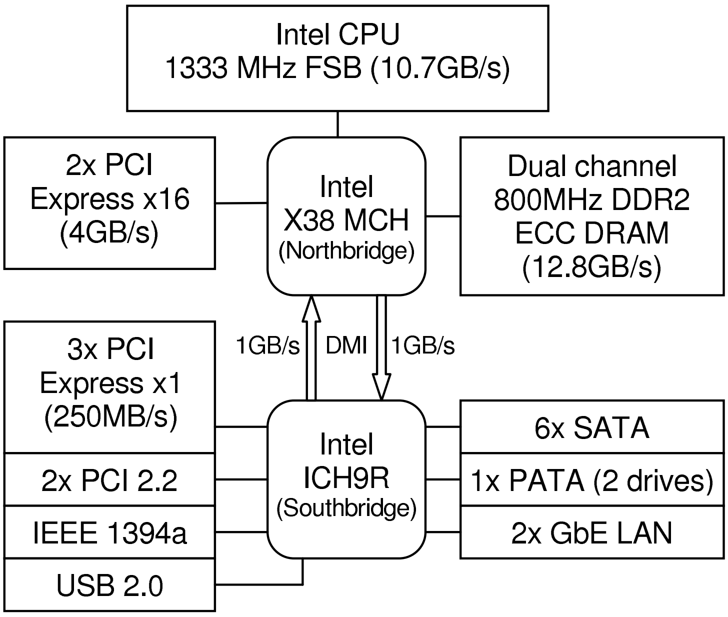
Intel CPU
1333 MHz FSB (10.7GB/s)
2x PCI Express x16 (4GB/s)
Intel X38 MCH (Northbridge)
Dual channel 800MHz DDR2 ECC DRAM (12.8GB/s)
3x PCI Express x1 (250MB/s)
1GB/s
DMI
1GB/s
Intel ICH9R (Southbridge)
2x PCI 2.2
IEEE 1394a
USB 2.0
6x SATA
1x PATA (2 drives)
2x GbE LAN

Figure 5: Overview of Asus Maximus Formula motherboard with Intel X38 chipset

3.11. Motherboard

After the HDD WTR, the motherboard is often the second most important bottleneck for data capture. Unfortunately, the actual performance of a motherboard is hard to predict, as it depends on a combination of many factors. But, first of all, it should have a sufficient number of storage connections, PCI-E slots, and memory capacity.

The most obvious choice is to use a high-end server motherboard, with a chipset such as the Intel 5000 or better, supporting only Intel Xeon CPUs. However, this may be more costly than necessary. Recently, the gaming industry has developed some consumer motherboards that are very well suited for video capture, and belong to the lower price range products.

Figure 5 shows the overview of the Asustek 'Maximus Formula' board, used in our experiments, that has an Intel X38 chipset. It supports up to 8GB of ECC DDR2 800MHz RAM and has 6 SATA connections (with RAID support), as well as the support for two Parallel-ATA (PATA) devices. This means that with 6 HDDs for image capture, a system disk and optical drive (for installing software) can still be connected to the PATA interface. The motherboard has two PCI-E×16 slots, that are connected directly to the

Northbridge, and three PCI-E×1 slots connected to the Southbridge.

During a video capture process, each Firewire Bus Card (FBC) transfers video data to DRAM, while the video capture application copies received video frames into DRAM frame buffers. From the frame buffers, the data is subsequently formatted (and possibly compressed) and transferred to the HDDs, connected to the Southbridge. The DMI link between North- and Southbridge limits the total HDD WTR to 1GB/s, minus overhead and other southbound data. The rate of northbound video data (coming from the FBCs) can be reduced by placing one or more of the FBCs in a PCI-E×16 slot (compatible with PCI-E×1, -×2 , -×4 and -×8), connected directly to the Northbridge.

When a PCI graphics card is used, five PCI-E×1 cards with dual IEEE 1394b bus can be installed, each of which supports 2×8 cameras. This totals to 740MB/s of video data from up to 80 cameras. Even more cameras could be connected through the on-board FireWire 400 and/or a PCI IEEE 1394b card.

Other consumer-class motherboards with similar specifications are the Asustek 'Rampage Formula' or 'P5E Deluxe' (which have the newer X48 chipset). The Gigabyte X38 or X48 boards are similar in functionality as well. Note, however, that there are reports of issues with audio recordings with these Gigabyte motherboards [22], related to high Deferred Procedure Call (DPC) latencies.

When we replaced the motherboard in our setup with the Gigabyte GA-EX58-UD5 (rev. 1.0, BIOS version F7), which has the more advanced X58 chipset, we regularly experienced a temporary audio dropout at the start of an A/V data capture process. This was solved by disabling 'hyper-threading' in BIOS. Hyper-threading has been re-introduced in the Intel Core i7 CPUs and provides a marginal increase in performance for some applications.

3.12. Software

Our proposed multi-sensor capture solution does not depend on the specific choice of software. However, when using COTS components, Microsoft Windows operating systems are currently the most suitable for multi-sensor applications. This is because the support of hard- and software for the mainstream consumer market is often solely aimed at these operating systems.

The video capture is handled by 'Streampix 4' [23], which can record video to HDD, from multiple sources simultaneously, and in a format that allows sequential disk writing. The latter is essential to reach the full WTR of a HDD. After the recording, the sequences can be processed, exported and compressed by any installed video CODEC.

When each sensor has its own capture software, controlling the starting, stopping and exporting of data recordings quickly becomes unmanageable. Unfortunately, many applications under MS Windows only work by Graphical User Interface (GUI), not allowing for scripting. This problem has been solved in the case of our system, by the freeware scripting package AutoIt v3, which can switch between applications, read window contents, activate controls and emulate keyboard and mouse actions.

4. System Throughput

The captured audio data consisted of 8 synchronous channels at 24-bit, 96kHz sampling rate. This amounts to only 2.2MB/s of data that was streamed to the HDD, which also contained the operating system and the software. Because the video data rates are orders of magnitude higher, and the data were streamed separately to the 6 SATA disks (see table 3), all our experiments concentrated on the video throughput. However, they were always conducted under the simultaneous audio capture.

The 8 FireWire cameras were not enough to test the capture system to full capacity. Therefore, we added 10 more GE1050 GbE cameras (as in table 4), set to capture a Region Of Interest (ROI) of 780x580pixels. The 8 FireWire cameras were connected through 2 PCI-E×1 dual FireWire cards on the southbridge chip of the motherboard. 2 of the GbE cameras were connected through the 2 motherboard LAN ports, also connected to the southbridge chip. The other 8 GbE cameras were connected through 2 PCI-E×4 quad network adapter cards (as in table 4), connected to the northbridge chip.

In paragraph 4.1 we present the results of testing the throughput of data storage, followed by the results of actual sensor data capture in paragraph 4.2. Paragraph 4.3 explains how a bottleneck in the system can be overcome, in order to capture more than double the amount of data.

4.1. Storage throughput results

To test the storage throughput of the system, we used the benchmarking tool '**HD_speed** v1.5.4.72'. One instance of **HD_speed** was used for each HDD, set to write with data blocks of 256kB. Figure 6 shows the WTR of writing to different numbers and configurations of HDDs simultaneously. These results show that the capacity of the SATA ports of the motherboard

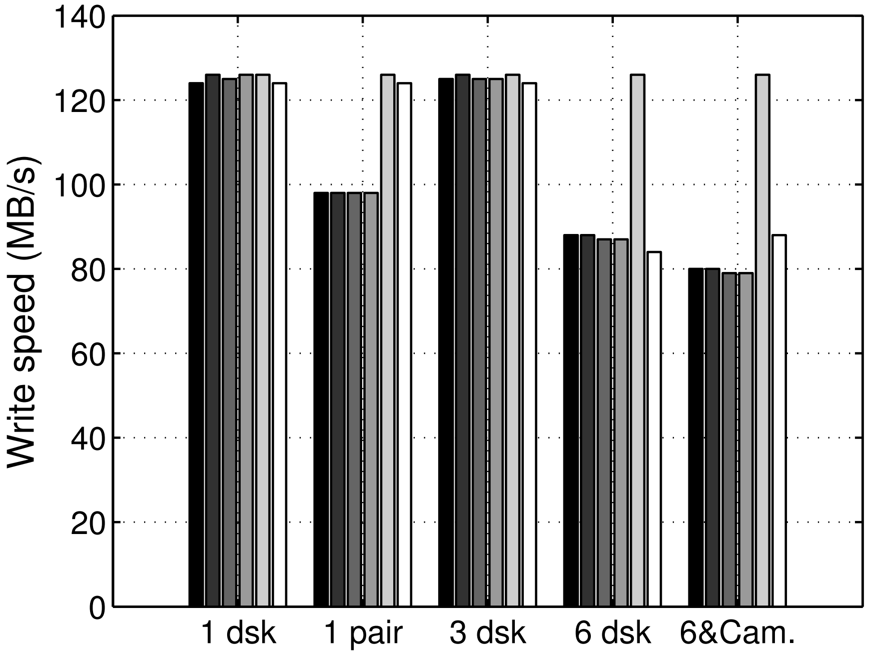

Figure 6: Sequential disk write performance of the 6 SATA ports on the motherboard (ICH9R controller) for different configurations. From left to right, the WTR of ports 1 to 6 are shown, shaded from black to white. '1 dsk': writing to 1 HDD at a time. '1 pair': writing to 2 disks simultaneously, through port 1&2, 3&4 or 5&6, respectively. '3 dsk': writing to 3 disks simultaneously, through port 1&3&5 or 2&4&6, respectively. '6 dsk': writing through all 6 ports at the same time. '6&Cam.': same as '6 dsk' but simultaneously streaming image data to memory, from 18 cameras of 780x580 pixels at 60fps.

are affected by each other, as well as by the incoming video data. The SATA ports hinder each other mostly in pairs (see '1 pair' in fig. 6), although the 5th SATA port is able to maintain the full 124MB/s of the HDD under all of the tested circumstances. Connecting 3 disks to SATA ports of different pairs ('3 dsk' in fig. 6) also allows to write at full HDD speed. When writing to all 6 HDDs, while simultaneously receiving video from 18 cameras at 60fps ('6&Cam.' in fig. 6), the minimum WTR to each separate disk was 79.3MB/s. This means that the system could store up to 475MB/s of data, with all disks writing at the same rate.

4.2. A/V capture throughput results

The maximum throughput of 475MB/s, found in paragraph 4.1, only holds for sequential writing from a single source in 256kB blocks. When writing video data from multiple sources (e.g. cameras) to a single HDD, the actual throughput may be lower. When streaming the data to HDD from the 18 cameras and the 8-channel audio interface at the same time, the temporal resolution of the cameras had to be limited to 40.1fps (313MB/s of data). Furthermore, to prevent the communication to the PCI graphics card from reducing the storage WTR, we had to disable displaying the live video. With 16 cameras, we could reach 49.9fps (346MB/s), and with 14 cameras we could reach the full camera frame rate of 61.7fps (375MB/s). The CPU load during these tests was around 70%.

When streaming the data from 3 cameras at 61.7fps (26.6MB/s per camera) to the same HDD, the data capturing could only run successfully up to 40% of HDD space. This is due to the reduction of WTR on the inner parts of the HDD platters (see figure 4). This means that, with 14 cameras at full speed, the usable storage size is only 571GB per disk, thus continuous capture is limited to two hours. With 12 cameras (2 cameras per HDD), the full disks can be used to record up to 7.6hours at 61.7fps.

4.3. Results after system upgrade

The above results indicate that the capture system has a bottleneck in the SATA controller of the motherboard. To be able to capture from 14 cameras with a resolution of 1024x1024pixels and 59.1fps, we made a few modifications, shown in table 4. The new GA-EX58-UD5 motherboard has more PCI-E slots connected to the northbridge chip (1 PCI-E×4 plus 3 PCI-E×16), while not using the northbridge for memory control anymore. Furthermore, we added an 8-port PCI-E×4 HDD controller card, together with

Table 4: Components of the modified capture system for 14 GigE Vision cameras with a resolution of 1024x1024pixels and 59.1fps

Sensor Component	**Description**
12 monochrome video cameras	Prosilica GE1050, 1024x1024 pixels resolution, max. 59.1fps
2 colour video cameras	Prosilica GE1050C, 1024x1024 pix. Bayer pattern, max. 59.1fps
3 quad port GbE Network cards	Intel PRO/1000 PT Quad-port PCI-E×4
HDD controller	Fujitsu Siemens RAID-CTRL SAS 8 Port PCI-E×4
2 SAS to SATA adapters	Adaptec Internal MSAS x4 To SATA
room microphone	AKG C 1000 S MkIII
head-worn microphone	AKG HC 577 L
external audio interface	MOTU 8-pre Firewire 8-channel, 24-bit, 96kHz

Computer Component	**Description**
14 Capture disks	Seagate Barracuda 1.5TB SATA, 32MB Cache, 7,200rpm
System disk	Samsung Spinpoint F1 1TB SATA, 32MB Cache, 7,200rpm
Optical drive	SATA DVD RW
6GB Memory	3x2GB 1600MHz DDR3 Corsair TR3X6G1600C7D
Graphics card	Matrox Millenium G450 16MB PCI
Motherboard	Gigabyte GA-EX58-UD5, ATX, Intel X58 chipset
CPU	Intel Core i7 920 S1366, 2.66GHz quad core, 8MB cache
Extended ATX Case	Thermaltake XASER VI
2 Cooled HDD enclosures	IcyBox Backplane System for 5x 3.5" SATA HDD
PSU	Akasa 1200W EXTREME POWER

Software Application	**Description**
MS Windows Vista 64 bit	64-bit Operating System
Norpix Streampix 4	Multi-camera video recording
Audacity 1.3.5	Freeware multi-channel audio recording
AutoIt v3	Freeware for scripting of Graphical User Interface control

8 extra SATA HDDs. Sequential WTR of this HDD controller was found to have a limit of 840MB/s, evenly distributed over all connected disks.

12 of the cameras were connected through 3 quad port PCI-E×4 network cards. 2 cameras were connected to the 2 internal LAN ports of the motherboard. Streaming to disk from all 14 cameras together with audio resulted in a system load of around 60% and was not affected by the displaying of live video. In this configuration, the total rate of the captured data is 830MB/s for a maximum recording duration of 6.7 hours.

5. Sensor Synchronisation

For many COTS sensor components, it is not possible to have external hardware control of the moments of data capture. When using software to synchronise data captured by different sensors, the synchronisation accuracy will be limited by the uncertainty in the latency between the sensor measurement and the handling of the data in the software. Depending on sensors, hardware and software, this latency may be anything from a few milliseconds up to more than hundreds of milliseconds. If there is no control over the exact sampling rates, synchronisation errors may even accumulate during a recording.

To synchronise between sensors, we centrally monitor the timings of all sensors, using the MOTU 8Pre external audio interface [24], connected to the capture PC through an IEEE 1394a connection. Since the analog inputs of the 8Pre are sampled using hardware-synchronised inputs (using the same clock signal), an event in one of the channels can be directly related to a temporal location in all other channels. The audio sampling rate determines the accuracy with which timing signals can be detected. The 8Pre can record up to 8 parallel channels at 24-bit, 96kHz. For our application, we used a sampling rate of 48kHz. This provides a 20μs granularity in determining signal timing.

In paragraph 5.1 we describe how to use this approach to synchronise sensors that have a trigger signal that can be externally measured. For sensors that do not have a measurable trigger signal, we describe how to accurately synchronise the PC system that captures the sensor data, in paragraph 5.2. In paragraph 5.3, we evaluate and discuss the synchronisation accuracy of eye gaze tracking data, synchronised without a trigger signal, followed by a discussion on sensor synchronisation in paragraph 5.4.

5.1. Synchronisation of sensors with a trigger signal

When a sensor has a measurable trigger signal (such as cameras that are externally triggered, or have a strobe output), this signal can be directly recorded alongside recorded sound, in a parallel audio track. Trigger voltages above the maximum input voltage of the audio interface can be converted with a voltage divider. The camera trigger pulses that we record in this way, can be easily detected and matched with all the captured video frames, using their respective frame number and/or timestamp. A rising camera trigger edge (see the 5th signal in figure 7) can be located in the audio signal with an accuracy of 1 audio sample. This means that, with an audio sampling rate of 48kHz, the uncertainty of localising the rising camera trigger edge is around 20μs. The frame exposures of the slave cameras start around 30μs later than the triggering camera, with a jitter of 1.3μs [17]. When this is taken into account, the resulting synchronisation error between audio and video can be kept below 25μs.

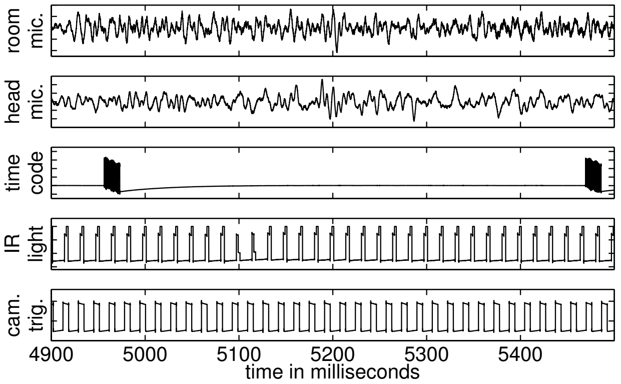

Figure 7: 5 tracks recorded in parallel by MOTU 8pre audio interface. From top to bottom: (1) room microphone; (2) head microphone; (3) serial port timestamp output (transmitted at 9600bps), showing 2 timestamp signals; (4) measured infrared light in front of eye tracker; (5) camera trigger.

Another advantage of this synchronisation method is that it allows the usage of COTS software applications for capturing each modality separately. Any type of sensor can be synchronised with the audio data, as long as it produces a measurable signal at the data capture moment, and its output data include reliable sample counts or timestamps relative to the first sample.

5.2. Synchronisation of sensors without a trigger signal

For sensors that do not have a trigger output, such as the Tobii X120 Eye Tracker, the synchronisation method described in paragraph 5.1 is not suitable. The data recorded by the eye tracker is timestamped using the CPU cycle counter of the computer that runs the Eye-tracker [25]. However, an additional procedure is required to relate a timestamp in local CPU time to the corresponding temporal location in the audio channels that are recorded with the MOTU 8pre in our setup. To establish this link, we developed an application which periodically generates and transmits timestamps of the momentary CPU cycle count time through the serial port. These timestamp signals are recorded in a separate audio track, in parallel to the microphone and camera trigger signals (see paragraph 5.1). Two examples of such a timestamp signal are shown in the 3rd audio track in figure 7.

A pair of the temporal start location of the timestamp signal in the audio recording, together with the time of the remote system retrieved by decoding the message in the timestamp signal, allows to relate this temporal location in all parallel recorded audio channels (e.g. sound, camera trigger pulses or timestamp signals from another PC) to the time of the remote sensorial data capture system. When two or more of such temporal pairs are known, the linear mapping can be determined that relates all temporal locations in the audio recording to timestamps of the sensorial data captured by the remote system.

We will first describe how the timestamp signals are generated using the serial port, followed by how they are extracted from the data recorded by the audio interface. Then, we describe how we use the recorded timestamps to find a linear time mapping between the computer system and the audio samples and show the results of applying this to recorded sequences.

5.2.1. Serial Port Timestamp Signal Generation

A standard serial port (RS-232 compatible interface) is used to generate a timestamp signal every 0.5 seconds, at a bit rate that can be easily read with the utilised audio interface. In our recordings, we used the MOTU8pre at 48kHz sampling rate and we configured the serial port to transmit at 9600 bits per second (bps). The output pin of the serial port is connected to the input pin. This allowed us to read back the transmitted timestamp to make an online estimate of the transmission latency, as described below. Each 16 byte long timestamp message consists of a concatenation of a marker pattern of 1 byte, two 4 byte numbers representing local time as a combination of seconds and microseconds, respectively, a 4 byte number representing the online prediction of the transmission latency in microseconds (which was applied to compensate the timestamp), 1 byte parity to detect a possible error in the message, and 2 bytes appended to obtain a message length that is divisible by 8. The marker pattern is an alternating bit pattern that is used to locate the start of a timestamp message by the procedure that reads back the transmitted timestamps.

Writing the generated timestamps to a serial port by a software application involves several steps that all take a certain amount of time to complete. The duration that the software application has to wait before the transmission command is completed depends on the speed of the system, as well as on other processes that may occupy the system for any amount of time. The time between writing the timestamp message to the port buffer and the commencing of the conversion of the message into an output signal, depends on the operating system architecture, the serial port hardware, as well as on the current state and settings of the hardware. All these latencies will cause a delay before the transmitted timestamp of the momentary local time is received by the audio input. If no compensation is provisioned, this will cause synchronisation inaccuracy. Therefore, we implemented an online estimation of the total transmission latency by reading back the serial port output directly into its input. Assuming that the process of transmission and reception are symmetric, the transmission latency can be found as half of the time needed for transmitting and receiving the timestamp signal, compensated by the duration of the signal. We use the running median of the estimated latencies of the last N transmissions as a prediction for the latency of the next transmission. The running median is robust against occasional extreme latencies, caused by other system processes that may block the transmission. The predicted latency is simply added to the timestamp, under the assumption that the timestamp will be exact at the moment of arrival.

A problematic issue inherent to this approach is that exact signal duration needs to be known in order to estimate the transmission latency (to be compensated by the signal duration). This proved to be impossible to achieve in a straightforward manner. We found out that the actual rate of transmission deviates slightly from to the specified bit rate, depending on the hardware. The difference was large enough to cause a significant deviation between the actual signal duration and the duration expected based on the message length and the specified bit rate. However, the actual bit rate of a specific serial port can be assumed to remain constant over time. Thus, it can be estimated beforehand by comparing the measured transmission times λ_1 and λ_2 of two messages of different bit lengths N_1 and N_2 (including start and stop bits), defined as follows:

$$\begin{aligned} \lambda_1 &= T_w + N_1/R + T_r \quad (1) \\ \lambda_2 &= T_w + N_2/R + T_r \quad (2) \end{aligned}$$

where R is the bit rate, and T_w and T_r relate to the (unknown) time needed to write to and read from the serial port buffer, respectively. Assuming symmetry, $T = T_w = T_r$, the bit rate R can be estimated as follows:

$$R = \frac{N_2 - N_1}{\lambda_2 - \lambda_1}. \quad (3)$$

The estimation of R from a sufficiently large number of measurements is used as an input parameter in our timestamp signal generator application.

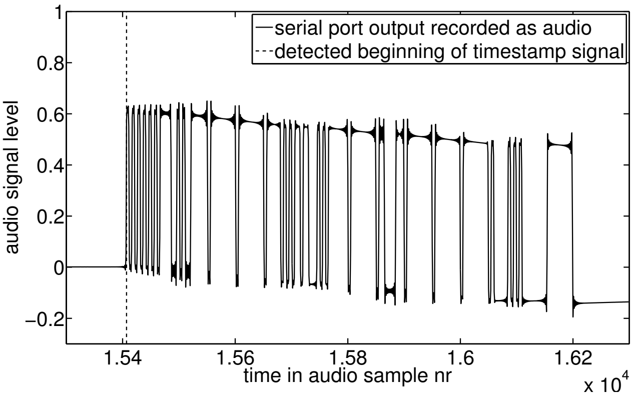

Figure 8: A binary (on/off) timestamp signal output from a serial port, recorded as an audio channel. A high-pass filter used in the audio processing causes vertical skew along the timestamp, while an anti-aliasing filter causes ripple around the step edges. The skew is compensated before reading the timestamp message, by a linear interpolation of the steady state level before and after the timestamp.

For the PC where the gaze tracker was running and which we synchronised with our A/V recordings, the measured transmission latency (for messages with a size that is a multiple of 8 bytes) was usually around 30μs. However, occasional outliers from this average can occur when transmission is interrupted by another system event. The largest outlier we came across during 7 hours of recording was around 25 milliseconds.

5.2.2. Timestamp Signal Processing

The binary (on/off) timestamp messages are extracted from the recorded audio signal by detecting the start and end moments of a message, and finding the transitions between the 'off' and 'on' level. Because of a high-pass filter used in the audio processing, the timestamp signal contains some vertical skew (see figure 8). This is compensated by interpolating the 'off' level according to the steady-state level before and after the timestamp signal.

5.2.3. Computer Synchronisation Evaluation

A pair of a detected onset moment of a timestamp signal in the recorded audio, together with the timestamp itself, can be used to relate the time in the audio recording to the time of the external system. Since hardware clocks in different systems do not run at (exactly) the same rate, one timestamp is

Table 5: Statistics of estimated root mean square errors (RMSE) in 87 recordings of approximately 5 minutes long, measured in number of Audio Samples (AS) at 48kHz or in μs. From left to right, this table shows the average, standard deviation, minimum and maximum of the RMSE estimated in 87 recordings.

	measure	**av. RMSE**	**σ RMSE**	**min RMSE**	**max RMSE**
(a)	Timestamp arrival vs. its linearisation	0.348AS / 7.25μs	0.546μs	6.51μs	9.38μs
(b)	IR pulse time vs. its linearisation	0.235AS / 4.90μs	1.03μs	3.44μs	9.74μs
(c)	Gaze data timestamp vs. IR pulse time	22.4AS / 467μs	287μs	122μs	1,443μs
(d)	Linearised gaze data vs. IR pulse time	15.2AS / 317μs	298μs	35.9μs	1.412μs

not enough to synchronise two systems. However, clocks that are driven by a crystal-oscillator (as is the case for practically all modern equipment), do run at a very constant rate. Therefore, we could find a linear mapping between audio sample number and the time of the external system, by applying a linear fit on all two-dimensional time synchronisation points (timestamps with corresponding audio time) that are received during a recording. To do so, we used linear regression with outlier exclusion. To have an idea about the consistency of individual timestamps, figure 9 shows the distribution of the time-difference of each individual timestamp compared the linear regression on all timestamps in one of our recordings. This shows that the timestamp signals were received and correctly localised within 1 audio sample (20μs) from the linear fit. Table 5 (a) shows statistics of the Root Mean Square Error (RMSE) (taking the linear fit as ground truth) over 87 recordings. In the RMSE measurements of the timestamps, we excluded the largest 1% of offsets (containing occasional extreme outliers). If we can assume that the latency compensation, described in paragraph 5.2.1, is unbiased, these results imply that an external system can be synchronised with an accuracy of approximately 20μs. The actual accuracy will depend on the linear regression method that is applied and on the length of a recording (the number of timestamp signals received).

5.3. Results for Synchronisation Without a Trigger Signal

In our experiments, the external system to be synchronised with the A/V data capture system was a PC running the Tobii X120 eye tracker. The Tobii X120 is connected to the PC by an ethernet connection. The Tobii Studio software package records the gaze tracking data with timestamps that are translated to the PC's local time, based on the CPU cycle counter. For

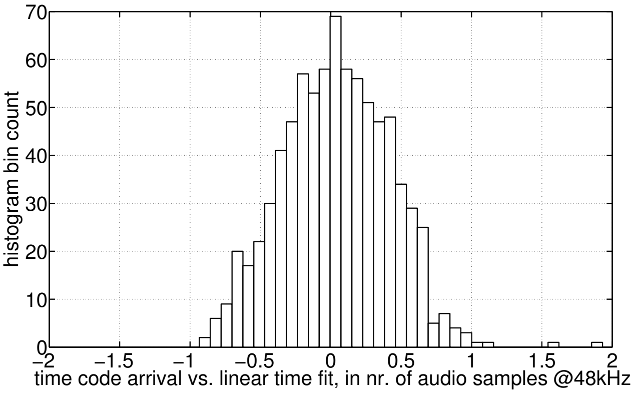

Figure 9: Histogram of the time differences between detected onsets of timestamp signals from a linear fit on all timestamps in one recording. A timestamp signal is based on the CPU cycle counter of an external PC, transmitted through its serial port and recorded as an audio signal at a sampling rate of 48kHz.

this translation, the clocks in the Tobii X120 and the PC are continuously synchronised by a protocol incorporated in the Tobii Studio software.

The Tobii X120 Eye tracker contains two cameras and two pairs of Infra Red (IR) light emitters of different type. The X120 has to rely completely on IR light, because the cameras are behind a filter glass that is opaque to visible light. The IR emitters are turned on during each image capture. Therefore, the moment of an IR flash should correspond to the moment of gaze data capture. Using a photo diode that is sensitive to IR, we could record these flashes as a sensor trigger signal in one of the audio channels and estimate the accuracy of synchronisation of the gaze data. Note that we cannot be sure that the IR light emissions correspond exactly to the data capture intervals, since this information about the working of the Tobii X120 is not provided. In any case, the data capture interval is limited by the IR emission intervals, since there is no light to capture without illumination. A data capture interval being (much) shorter than the IR emission would be unlikely, since the emitted light is already scarce due to the limited maximum power of the emitters, as well as due to the safety regulations imposed on the exposure of the human eye to the IR light.

An example of the comparison between the timestamps of the captured gaze data and the IR flashes is shown in figure 10. Note that, for our ex-

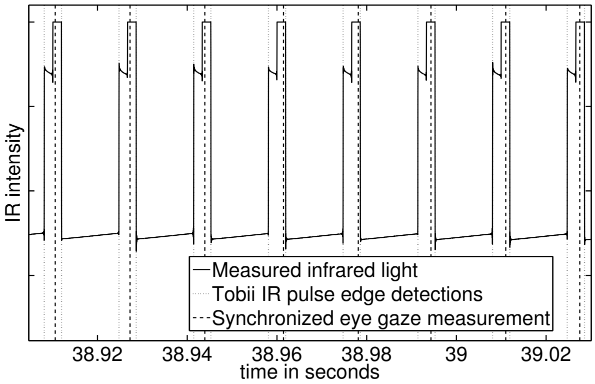

Figure 10: Comparison between infrared light flashes from the Tobii X120 eye tracker (set to 60Hz) and the timestamps of the recorded gaze data. The infrared light, measured by a photo diode in front of the Tobii X120, is recorded by an audio interface at 48kHz. In this fragment, the largest deviation between the centre of the time interval of the IR flash and its corresponding data timestamp is 1.46ms.

periments, we have set the Tobii X120 to 60Hz rather than 120Hz, because this allows more freedom of head movement [25]. The timestamps assigned to the gaze data by the Tobii Studio software corresponded mostly to the middle of the time interval of the IR flashes.

Apart from a few outliers, the IR flashes showed a high temporal regularity. Figure 11 shows the distribution of the time-difference of each individual estimated centre of an IR flash time interval compared to a linear fit to all centres, for one of our recordings. The majority of the flashes is located within 0.5 audio samples (10μs at 48kHz) from the linear fit. Table 5 (b) shows statistics of the Root Mean Square Error (RMSE) measure over 87 recordings, with a worst-case RMSE being 5.96μs. Besides the temporal regularity of the IR flashes, this also suggests that the localisation of flash moments is reliable and that the audio sampling rate of the audio interface is constant.

Assuming that the centres of the time intervals of the IR flashes are the actual moments of gaze data capture, and that each gaze datum and its nearest IR flash correspond to each other, we can evaluate the accuracy of the gaze data timestamps after converting them to the corresponding time in the audio recording using the linear mapping described in paragraph 5.2.3. Figure 12 shows the progression of the estimated gaze data timestamp error

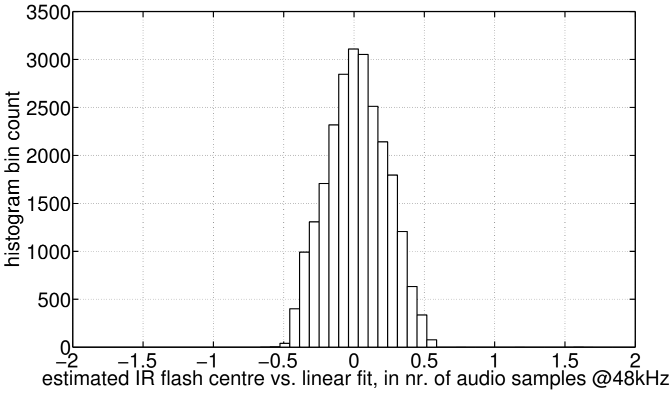

Figure 11: Histogram of the difference of the estimated time interval centres of IR flashes, compared to their linear fit. Flashes were recorded with a photo diode connected to an audio input and placed in front of the Tobii X120 eye tracker.

over time, for one of our recordings. In contrast to the high temporal regularity of the IR flashes, the timestamps of the captured gaze data show highly irregular differences with the IR flash moments. Since the synchronisation between the PC and the audio interface is linear over the entire recording, the only possible sources of these irregularities can be an inconsistent latency in the LAN connection between the Tobii X120 and the PC, or a variation in how long it takes before the incoming data is processed by Tobii Studio. Table 5 (c) shows statistics of the RMSE over 87 recordings and figure 13 shows the distribution of the gaze data timestamp errors over all recordings. We have excluded data samples for which one half of the expected IR flash interval was missing. We could not be sure about the flash interval centre for these cases; thus we had no baseline to determine the error.

The largest error we measured overall was 3.6 milliseconds. This means that the timestamp of a gaze datum can be corrected by the closest IR flash interval centre, localised with an accuracy of 0.5 audio samples (10μs at 48kHz).

Knowing that the Tobii X120 records the data at regular intervals, a straightforward way to improve the accuracy of assigned timestamps (without recording the IR pulses) is by fitting a linear function directly to the gaze data timestamps. The result of this correction for the first recording is shown as the gray line in figure 12. The related statistics for the RMSE are shown

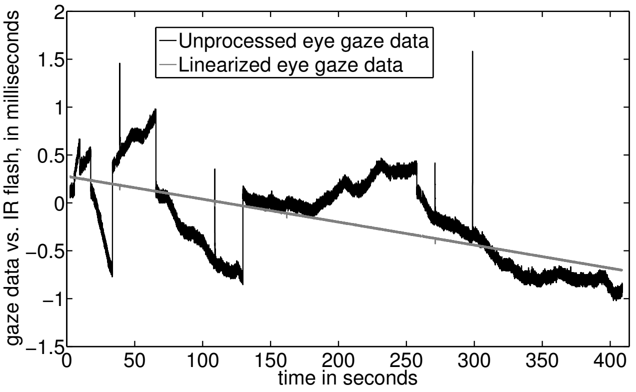

Figure 12: Estimated gaze data timestamp error over time, in comparison to the interval centre of the closest IR flash, measured in the audio recording.

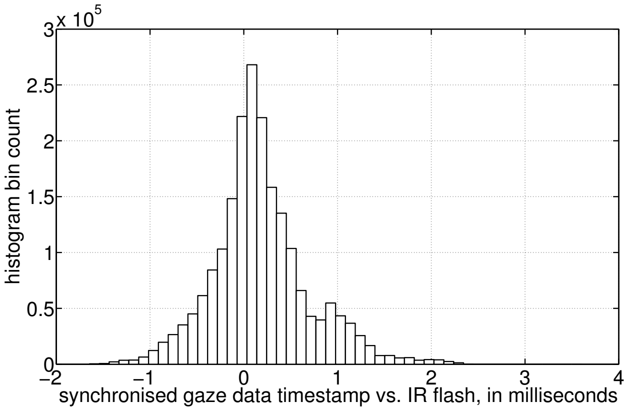

Figure 13: Histogram of the difference between a gaze data timestamp and the time interval centre of the corresponding IR flash, measured over 87 recordings of approximately 5 minutes long. The most extreme offset measured among these 2028526 data samples was 3.60milliseconds.

in table 5 (e). Linearising led to an overall improvement of around 32%. Although the amount of improvement varied a lot per recording, it led to a lower RMSE in all cases. In longer recordings, the benefit of linearising the timestamps will be more significant.

5.4. Discussion on Synchronisation

Capture software running on different PCs can be synchronised by letting each PC transmit its CPU cycle count as timestamp signals outputted by the serial port. The timestamp signals from multiple PCs can be recorded as separate channels in a multi-channel audio interface, making use of the hardware-synchronisation between the different audio channels. Furthermore, Radio Frequency (RF) transmission of these timestamp signals allows for wireless integration of various systems [5]. And since the same timestamp signal can be connected to multiple audio interfaces, it also allows straightforward expansion of the number of synchronised audio channels, beyond the capacity of any single audio interface.

The above-discussed experiments show that synchronisation by transmitting timestamp signals through the serial port, can be done with an accuracy of approximately 20μs. However, the exact accuracy depends on various delays of sensor measurements, data recordings and synchronisation between sensor-hardware and the CPU cycle count of the PC that captures the data. The example of the Tobii X120 eye tracker demonstrates that the synchronisation of two data capture systems is not a trivial matter. When synchronising captured data with data captured by another system, one has to make sure that the data has been captured with sufficient accuracy in the first place. Therefore, in order to avoid relying on synchronisation protocols with insufficient, uncontrollable, or unknown uncertainty, it is recommendable to use sensors with a measurable trigger signal.

6. Conclusions

We have proven that it is possible to build a complete solution for multi-camera and multi-sensor data capture, with accurate synchronisation between modalities and systems, using only Commercial Off-The-Shelf (COTS) hardware components. Our approach does not require complicated or expensive synchronisation hardware, and allows the usage of separate capture software for each modality, maximising flexibility with minimal costs.

For sensor synchronisation, we have proposed to use a multi-channel audio interface to record audio alongside the trigger signals of externally triggered sensors. For sensors without an external trigger signal, we have presented a method to generate timestamp signals with a serial port, allowing to synchronise a PC that captures sensor data. Experiments show that the resulting synchronisation of a CPU cycle counter is accurate within 20μs. In practice, however, synchronisation will be limited by jitter and uncertainty in latencies in the actual sensor hard- and software that is used. Synchronised eye gaze data from a Tobii X120 eye tracker, showed errors up to 3.6 milliseconds. Because the data was recorded at 60Hz (with 16ms intervals), we could use the infrared light pulses, emitted during data capture of the Tobii X120 and measured with a photo diode, to correct the errors up to 10μs accurate.

Using low-cost COTS components, we built an audio/video capture PC that was capable of capturing 7.6 hours of video simultaneously from 12 cameras with resolutions of 780x580 pixels each, at 61.7 fps, together with 8 channels of 24-bit audio at 96kHz sampling rate. When capturing from 18 cameras, a bottleneck in the southbridge chip of the system's motherboard limited the frame rate to 40.1fps. Using a motherboard with more high-bandwidth PCI-E slots connected to the northbridge chip, together with a PCI-E×4 HDD controller for 8 extra HDDs, we were be able to record 8 channels of audio together with the video from 14 GigE Vision cameras of 1024x1024pixels at 59.1fps, for a duration of 6.7 hours. The captured data rate of this configuration amounts to a total of 830MB/s.

Acknowledgment

The research of M. Valstar leading to these results is funded in part by the European Community's 7th Framework Programme [FP7/2007-2013] under the grant agreement no 211486 (SEMAINE). The research of the other authors is funded by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB).

References

[1] A. Vinciarelli, M. Pantic, H. Bourlard, Social signal processing: Survey of an emerging domain, Image and Vision Computing (to appear) (2009) .

[2] H. Gunes, M. Pantic, Automatic, dimensional and continuous emotion recognition, International Journal of Synthetic Emotions 1 (1) (2010, in press) .

[3] K. W. Grant, V. van Wassenhove, D. Poeppel, Discrimination of auditory-visual synchrony, in: International Conference on Audio-Visual Speech Processing, 2003, pp. 31–35.

[4] M. Sargin, Y. Yemez, E. Erzin, A. Tekalp, Audio-visual synchronization and fusion using canonical correlation analysis, IEEE Transactions on Multimedia 9 (7) (2007) 1396–1403.

[5] R. Lienhart, I. Kozintsev, S. Wehr, Universal synchronization scheme for distributed audio-video capture on heterogeneous computing platforms., in: The Eleventh ACM international Conference on Multimedia, ACM, Berkeley, CA, USA, 2003, pp. 263–266.

[6] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, in: ACM SIGGRAPH, Los Angeles, CA, USA, 2004, pp. 600–608.

[7] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, M. Levoy, High performance imaging using large camera arrays, ACM Trans. Graph. 24 (3) (2005) 765–776.

[8] S. Tan, M. Zhang, W. Wang, W. Xu, Aha: An easily extendible high-resolution camera array, in: Second Workshop on Digital Media and its Application in Museum & Heritages, IEEE, 2007, pp. 319–323.

[9] T. Svoboda, H. Hug, L. Van Gool, Viroom - low cost synchronized multicamera system and its self-calibration, in: Proceedings of the 24th DAGM Symposium on Pattern Recognition, Vol. 2449 of Lecture Notes In Computer Science, Springer-Verlag, London, UK, 2002, pp. 515 – 522.

[10] X. Cao, Y. Liu, Q. Dai, A flexible client-driven 3dtv system for real-time acquisition, transmission, and display of dynamic scenes, EURASIP Journal on Advances in Signal ProcessingArticle No. 5.

[11] T. Hutchinson, F. Kuester, K.-U. Doerr, D. Lim, Optimal hardware and software design of an image-based system for capturing dynamic

movements, IEEE Transactions on Instrumentation and Measurement 55 (1) (2006) 164 – 175.

[12] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, Y. Suenaga, Multipoint measuring system for video and sound - 100-camera and microphone system, in: IEEE International Conference on Multimedia and Expo, 2006, pp. 437 – 440.

[13] P. Ekman, W. V. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologist Press, Palo Alto, CA, 1978.

[14] P. Ekman, W. V. Friesen, J. C. Hager, Facial Action Coding System, Research Nexus eBook, Salt Lake City, UT, 2002.

[15] H.-J. Freund, H. Bdingen, The relationship between speed and amplitude of the fastest voluntary contractions of human arm muscles, Journal of Experimental Brain Research 31 (1) (1978) 1–12.

[16] A. El Gamal, Trends in cmos image sensor technology and design, in: International Electron Devices Meeting, 2002, 2002, pp. 805–808.

[17] Allied Vision Technologies GmbH, Taschenweg 2a, D-07646 Stadtroda, Germany, AVT Stingray Technical Manual V4.2.0 (28 May 2009).

[18] S. Sookman, Choosing a camera interface: qualify and quantify, Advanced Imaging.

[19] A. N. Bashkatov, E. A. Genina, V. I. Kochubey, V. V. Tuchin, Optical properties of human skin, subcutaneous and mucous tissues in the wavelength range from 400 to 2000 nm, J. Phys. D: Appl. Phys. 38 (2005) 2543–2555.

[20] T. Tajbakhsh, R. Grigat, Illumination flicker correction and frequency classification methods, in: R. Martin, J. DiCarlo, N. Sampat (Eds.), Digital Photography III, Vol. 6502, SPIE, 2007, p. 650210.

[21] O. Hoshuyama, A. Sugiyama, A. Hirano, A robust adaptive microphone array with improved spatial selectivity and its evaluation in a real environment, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97), Vol. 1, 1997, p. 367.

[22] Anonymous, Gigabyte Boards and DPC Latency, Anandtech Forum, http://forums.anandtech.com/messageview.aspx?catid=29&threadid=2182171 (April 2008).

[23] Norpix Streampix 4 product description, http://norpix.com/products/multicamera.php (2007).

[24] MOTU 8pre product description, http://www.motu.com/products/motuaudio/8pre (2008).

[25] Tobii Technology AB, User Manual: Tobii X60 & X120 Eye Trackers, Revision 3 (November 2008).