Image and Vision Computing xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Cost-effective solution to synchronised audio-visual data capture using multiple sensors[☆]

Jeroen Lichtenauer [a,*], Jie Shen [a], Michel Valstar [a], Maja Pantic [a,b]

[a] Department of Computing, Imperial College London, 180 Queen's Gate, SW7 2AZ, UK
[b] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands

ARTICLE INFO

Article history:
Received 13 February 2011
Received in revised form 7 June 2011
Accepted 18 July 2011
Available online xxxx

Keywords:
Video recording
Audio recording
Multisensor systems
Synchronisation

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 due to several challenges, such as dealing with the combined data rate from multiple sensors; unknown 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 time-stamping 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 1024×1024 spatial- and 59.1 Hz temporal resolution, from at least 14 cameras, together with 8 channels of 24-bit audio at 96 kHz. We thus improve the quality/cost ratio of multi-sensor systems data capture systems.

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 200 ms or a video lag of up to 45 ms [3], multimodal data fusion algorithms may benefit from higher synchronisation accuracy. For example, in [4], correction of a 40 ms 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 [6]). 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].

[☆] This paper has been recommended for acceptance by Jan-Michael Frahm.
[*] Corresponding author. Tel.: +44 20 7594 8336; fax: +44 20 7581 8024.
E-mail address: j.lichtenauer@imperial.ac.uk (J. Lichtenauer).

0262-8856/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2011.07.004

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 [7] builds custom solutions for any application, and Cepoint Networks offers professional video equipment such as the Studio 9000™ DVR [8], 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 time-stamping 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 780 × 580 pixels spatial resolution and 61.7 fps temporal resolution, together with eight 24-bit 96 kHz audio channels. The relevant components of our system setup are summarised in Table 1. By using six internal 1.5 TB Hard Disk Drives (HDD), 7.6 h 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 1024 × 1024 pixels spatial resolution and 59.1 fps, for up to 6.7 h. In Table 2 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.

Table 1
Components of the capture system for 8 FireWire cameras with a resolution of 780 × 580pixels and 61.7 fps.

Sensor component	Description
7 monochrome video cameras	AVT Stingray F-046B, 780 × 580 pixels resolution, max. 61.7 fps
Colour video camera	AVT Stingray F-046C, 780 × 580 pix. Bayer pattern, max. 61 fps
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 577L
External audio interface	MOTU 8-pre FireWire 8-channel, 24-bit, 96 kHz
Eye tracker	Tobii X120
Computer component	**Description**
6 capture disks	Seagate Barracuda 1.5 TB SATA, 32 MB Cache, 7200 rpm
System disk	PATA Seagate Barracuda 160 GB 2 MB Cache, 7200 rpm
Optical drive	PATA DVD RW
4 GB Memory	2 GB PC2-6400 DDR2 ECC KVR800D2E5/2 G
Graphics card	Matrox Millenium G450 16 MB PCI
Motherboard	Asus Maximus Formula, ATX, Intel X38 chipset
CPU	Intel Core 2 Duo 3.16 GHz, 6 MB Cache, 1333 MHz 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 and stimuli data suite
Tobii SDK	Eye tracker Software Development Kit

Table 2
Camera support of a single consumer PC.

Spatial resolution	Temporal resolution	Rate per camera	Max. no. of cameras
780 × 580 pixels	61.7 fps	26.6 MB/s	14
780 × 580 pixels	49.9 fps	21.5 MB/s	16
780 × 580 pixels	40.1 fps	17.3 MB/s	18
With controller card for 8 additional HDDs			
1024 × 1024 pixels	59.1 fps	59.1 MB/s	14

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 time-stamping 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. A system overview is shown in Fig. 1. For sensors with an external trigger (b), we record the binary trigger signals directly into a separate audio track, parallel to tracks with recorded sound. For sensors that don't have an external trigger signal (f), we let the computer that captures the sensor data (e) periodically generate binary 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 six parts. We begin with describing related multi-camera capture solutions that have been proposed before, in Section 2. In Section 3, we describe important choices that need to be made for sensors that will be used in a multi-

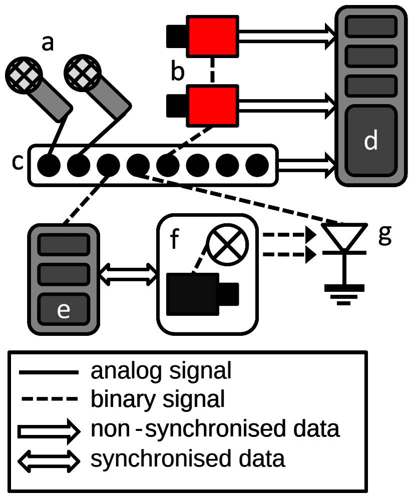

Fig. 1. Overview of our synchronised multi-sensor data capture system, consisting of (a) microphones, (b) video cameras, (c) a multi-channel A/D converter, (d) an A/V capture PC, (e) an eye gaze capture PC, (f) an eye gaze tracker and (g) a photo diode to capture the pulsed IR-illumination from (f).

Please cite this article as: J. Lichtenauer, et al., Cost-effective solution to synchronised audio-visual data capture using multiple sensors, Image Vis. Comput. (2011), doi:10.1016/j.imavis.2011.07.004

sensor data capture system. The following three sections cover three different problems of synchronised multi-sensor data capture: achieving a high data-throughput (Section 4), synchronisation at sensor-level (Section 5) and synchronisation at computer-level (Section 6), respectively. Each section describes how we have solved the respective problem and presents experimental results to evaluate the resulting performance. Finally, Section 7 contains our conclusions about the achieved knowledge and improvements.

2. Related multi-sensor video capture solutions

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. [9] used two specially built concentrator units to capture video from eight cameras of 1024×768 pixels spatial resolution at 15 fps.

Wilburn et al. [10] built an array of 100 cameras, using 4 PCs and custom-built low-cost cameras of 640×480 pixels spatial resolution at 30 fps, 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 ns between subsequent levels of the tree. For a tree of 100 cameras, this should result in a frame time difference of 1.2 μs, between the root and the leaf nodes.

More recently, a modular array of 24 cameras (1280×1024 pixels at 27 fps) was built by Tan et al. [11]. 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. [12] 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 640×480 pixels spatial resolution at 30 fps, although their software was limited to handling a temporal resolution of 10 fps. 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 [13], which could handle 4 cameras of 640×480 pixels spatial resolution at 30 fps per PC. The synchronisation accuracy between cameras was reported to be within 15 ms.

Hutchinson et al. [14] 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 658×494 pixels spatial resolution at 80 fps.

Fujii et al. [15] have developed a large-scale multi-sensorial setup capable of capturing from 100 cameras of 1392×1040 pixels spatial resolution at 29.4 fps, as well as from 200 microphones at 96 kHz. 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 analogue trigger signals from the same clock. By using a single, centralised trigger source for all measurements, the synchronisation error between sensors is kept below 1 μs. Disadvantages of this system are the high cost and volume of the equipment, as well as the required custom built hardware.

Table 3 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 an exact 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 between audio and video, as well as with other sensors and computer systems.

Table 3
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. [9] it was a 'concentrator unit'. 'cam.#/unit' indicates the maximum number of cameras that can be connected to a unit, 'audio#/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 type or accuracy of synchronisation between cameras and 'A/V sync' the accuracy of synchronisation between audio and video.

Solution	Cam.#/unit at 640×480 30 fps	Audio #/unit	Sync unit#	Unit sync	Camera sync	A/V sync
Our solution	14×1024×1024 p at 59.1 fps	<7	Unlimited	<20 μs	~30 μs	<25 μs
Studio 9000 DVR	4	Optional via SDI	Unlimited with IRIG-B	Optional IRIG-B	Depends on the cameras	Not specified
typical CCTV DVR	16	16	1	n.a.	Depends on the cameras	Not specified
Zitnick et al. [9]	4×1024×768 p at 15 fps	n.a.	≥2 (not specified)	By FireWire	Not specified	n.a.
Wilburn et al. [10]	30	n.a.	Unlimited	Hardware trigger	1.2 μs with 100 cameras	n.a.
Tan et al. [11]	1×1280×1024 p at 27 fps	n.a.	Unlimited	Hardware Trigger	Hardware Trigger	n.a.
Svoboda et al. [12]	3 at 10 fps	n.a.	Unlimited	Network trigger	Software trigger	n.a.
Cao et al. [13]	4	n.a.	Unlimited	15 ms with 16 units	Software trigger	n.a.
Hutchinson et al. [14]	4×658×494 p at 80 fps	n.a.	1	n.a.	Software trigger	n.a.
Fujii et al. [15]	1×1392×1040p at 29.4 fps	4	Unlimited	<1 μs with 100 units	<1 μs with 100 cameras	<1 μs with 100 units

Please cite this article as: J. Lichtenauer, et al., Cost-effective solution to synchronised audio-visual data capture using multiple sensors, Image Vis. Comput. (2011), doi:10.1016/j.imavis.2011.07.004

3. Sensor- and measurement considerations

Having a cost-saving and accurate sensor-synchronisation method is only relevant if it is combined with a set of recording equipment that is equally cost-effective and suited for the intended purpose. The quality of the captured data is limited by the quality of the sensors and is interdependent with the data-bandwidth, the synchronisation-possibilities as well as the quality of the recording conditions. Many important considerations were not familiar to us before we started building our recording setup. To facilitate a more (cost-) effective and smooth design process, this section covers the most important aspects that we had to take into consideration in choosing audio-visual sensors for a human behaviour analysis application.

We start by covering several important aspects of illumination, followed by the most important camera properties and image post-processing procedures. Subsequently, we discuss different microphone options for recording a person's vocal sounds and some comments on the use of COTS software. This section ends with an example of the costs we spent on equipment regarding each of these different aspects.

3.1. Illumination

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

3.1.1. 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 aberration. Most monochrome cameras are sensitive to the Near Infra Red (NIR) wavelengths (between 700 nm and 1000 nm). 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 [16]. This has a smoothing effect on wrinkles, irregularities and skin impurities, which can be beneficial to some applications of computer vision.

Note that incandescent studio lights often have a strong infrared component that can interfere with active infrared sensors. The Tobii gaze tracker we discuss in Section 6.4 was adversely affected by a 500 Watt incandescent light, while it worked well with two 50 Watt white-light LED arrays that produce a comparable brightness.

3.1.2. Illumination intensity

The intensity of light cast on the target object will determine the trade-off between shutter time and image 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).

3.1.3. 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.

3.1.4. 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.

3.1.5. 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 60 Hz) 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 100 Hz period, the minimum shutter time is 10 ms. 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. [17]) 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.

3.1.6. 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 s, 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 (DoF), 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.

3.2. 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 250 ms[18,19], and that the fastest hand movements (finger movements) last 80 ms[20]. Hence, in order to

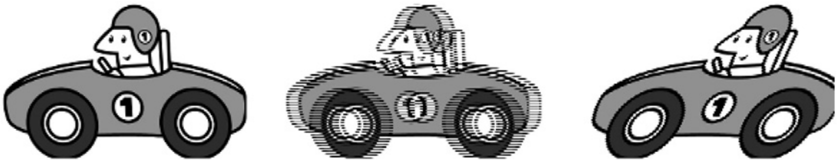

Fig. 2. Example of how an image of a horizontal moving object looks like, when captured with a camera with (1) progressive scan with global shutter (left), (2) interlaced scan (middle) and (3) progressive scan with rolling shutter (right).

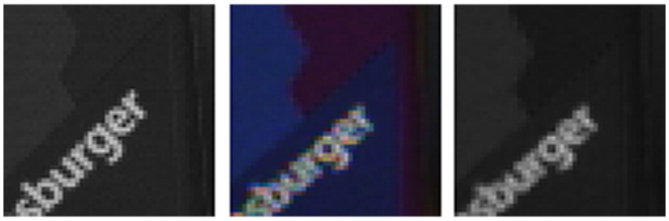

Fig. 3. Comparison of the AVT Stingray F-046B monochrome camera with shutter 1/60s (left) to the AVT Stingray F-046C Bayer colour camera with shutter 1/20s (middle). The right image is obtained by converting the colour image to a grey image.

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. Fig. 4 shows a fast head turn captured at 60 fps.

3.3. 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 Fig. 2. 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.

3.4. 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). Fig. 3 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 Fig. 3).
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' [21]. 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.

3.5. 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 DoF and/or shutter time. Fig. 4 shows the effects of the trade-off between

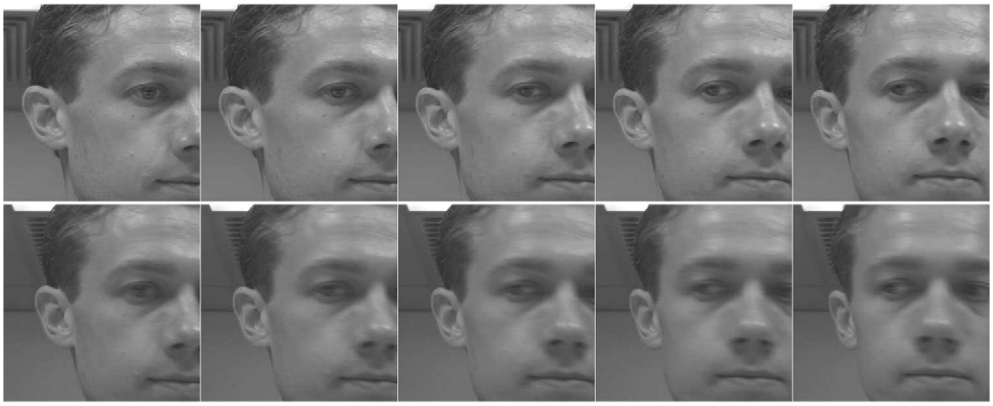

Fig. 4. Example of two trade-offs between shutter time and aperture. The recorded action is a quick head turn as the result of a sudden change of attention. The images are cropped at 300 × 300 pixels from a full resolution of 780 × 580 pixels. The top row shows 5 subsequent images taken at 60 fps, with a shutter time of 5 ms. The bottom row shows images taken at the same moments, from a synchronised camera, with a shutter time of 15 ms and a smaller aperture, to obtain the same image brightness. The result of the longer shutter time is an increased motion blur, while the smaller aperture results in a sharper background due to the increased DoF.

Please cite this article as: J. Lichtenauer, et al., Cost-effective solution to synchronised audio-visual data capture using multiple sensors, Image Vis. Comput. (2011), doi:10.1016/j.imavis.2011.07.004

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.

3.6. 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 [22]. 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 30 μs. 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 30 μs delay is added.

3.7. 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 [23]. 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 (74 MB/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 100 MB/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 more difficult 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 [23], which can be important in time-critical applications where a system has to respond with low latency.

For our application that requires cameras with a spatial resolution of 780 × 580 pixels and a temporal resolution of 60 fps (25.9 MB/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 requires a resolution of 1024 × 1024 pixels and 60 fps (60 MB/s), we chose for the GigE Vision interface. With 60 MB/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 doesn't pose a problem in our setup.

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 aberration 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 quality-to-size ratio of contemporary multi-pass compression methods (e.g. H.264) is significantly higher than what can currently 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 [24].

3.10. Sensor capture software

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

The video capture is handled by 'Streampix 4' [25], 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-for-Windows 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-W) 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.

3.11. Equipment costs

Table 4 lists the costs we have spent on a setup to record a person's responses when sitting in front of a digital screen showing media fragments. A detailed description of this experiment and the available database of the recordings can be found in [26]. The equipment included six cameras and two microphones, similar to those in Table 1. We chose to illuminate the person with two 40 Watt LED arrays. Such lights are cool enough to be used at a short distance without causing physical discomfort and have an intensity that is stable enough to be used with exposure durations well below 10 ms (the limit for lights directly powered by a 50 Hz AC source).

Table 4
Equipment costs for an experimental setup using the proposed synchronisation method. Costs excl. VAT are in pound sterling (GBP) and include required accessories such as cables and tripods.

Equipment	Details	Costs in GBP(£)
Data capture PC	Includes video capture disks	£1500
Audio hardware	Two mics and an 8-channel input	£1200
Illumination	Two 40 W LED arrays	£2500
Cameras	6 cameras and 6 lenses	£5000
Software	For multi-camera recording	£1000
Total		£11,200

4. High-throughput data capture

In multi-sensor data capture, it is crucial to have a sufficient throughput to capture from all sensors simultaneously. Maximising the amount of sensors that can be handled by one computer does not only reduce costs, but also improves a system's space requirements as well as its ease of setup and use.

In Sections 4.1 and 4.2, we describe the motivations behind the most important choices we had to make for the storage hardware and the motherboard, respectively, in order to obtain sufficient system capacity. Experiments and results for the evaluation of our system's capacity are presented in Section 4.3.

4.1. Storage

Currently, the Hard Disk Drive (HDD) is the most significant bottleneck of a conventional PC. Capturing to Dynamic Random Access Memory (DRAM) is the best solution for short video fragments. However, many applications require significantly longer recordings than what can be stored in DRAM. The fastest consumer HDDs, with Serial Advanced Technology Attachment (SATA) interface, currently start with a data rate of over 100 MB/s (at the outside of the platter) and gradually descend to a rate of around 60 MB/s at the end of the disk. The decrease in Write Transfer Rate (WTR) of a 1.5 TB Seagate Barracuda disk is shown in Fig. 5.

Most high-end consumer motherboards provide SATA connections for six disks, including hardware RAID support, which will allow a total capture rate of approximately 500 MB/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.

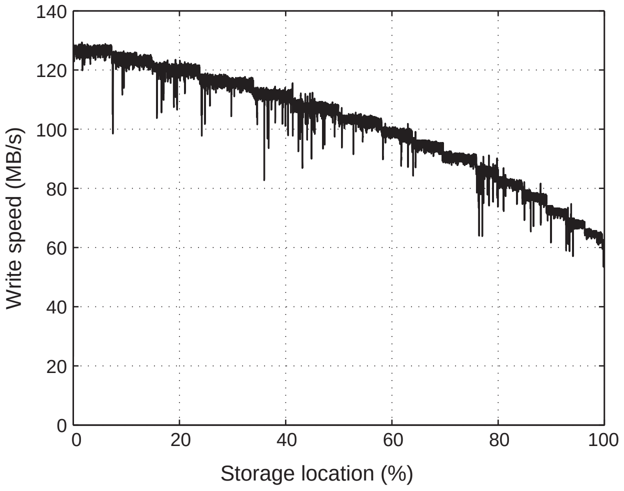

Fig. 5. Sequential write transfer rate of 1.5 TB Seagate Barracuda HDD as a function of disk location.

Please cite this article as: J. Lichtenauer, et al., Cost-effective solution to synchronised audio-visual data capture using multiple sensors, Image Vis. Comput. (2011), doi:10.1016/j.imavis.2011.07.004

4.2. 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.

Fig. 6 shows the overview of the Asustek 'Maximus Formula' board, used in our experiments, that has an Intel X38 chipset. It supports up to 8 GB of ECC DDR2 800 MHz DRAM 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 1 GB/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 740 MB/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 [27], 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.

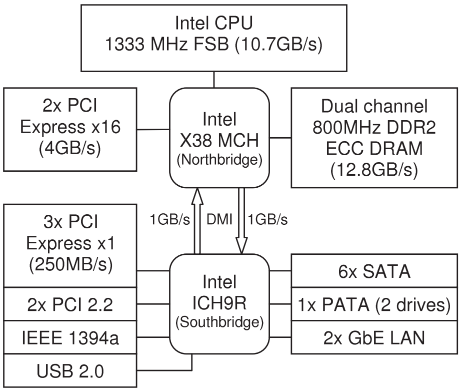
Intel CPU
1333 MHz FSB (10.7GB/s)
2x PCI Express x16 (4GB/s)
Intel X38 MCH (Northbridge)
Dual channel 800MHz DDR2 ECC DRAM (12.8GB/s)
1GB/s
DMI
1GB/s
3x PCI Express x1 (250MB/s)
2x PCI 2.2
IEEE 1394a
USB 2.0
Intel ICH9R (Southbridge)
6x SATA
1x PATA (2 drives)
2x GbE LAN

Fig. 6. Overview of Asus Maximus Formula motherboard with Intel X38 chipset.

4.3. System throughput experiments

The captured audio data consisted of 8 synchronous channels at 24-bit, 96 kHz sampling rate. This amounts to only 2.2 MB/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 1), 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 5), set to capture a Region Of Interest (ROI) of 780 × 580pixels. 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 5), connected to the northbridge chip.

In Section 4.3.1 we present the results of testing the throughput of data storage, followed by the results of actual sensor data capture in Section 4.3.2. Section 4.3.3 explains how a bottleneck in the system can be overcome, in order to capture more than double the amount of data.

4.3.1. 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

Table 5
Components of the modified capture system for 14 GigE Vision cameras with a resolution of 1024 × 1024pixels and 59.1 fps.

Sensor component	Description
12 monochrome video cameras	Prosilica GE1050, 1024 × 1024 pixels resolution, max. 59.1 fps
2 colour video cameras	Prosilica GE1050C, 1024 × 1024 pix. Bayer pattern, max. 59.1 fps
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 ×4 to SATA
Room microphone	AKG C 1000 S MkIII
Head-worn microphone	AKG HC 577L
External audio interface	MOTU 8-pre FireWire 8-channel, 24-bit, 96 kHz
Computer component	Description
14 capture disks	Seagate Barracuda 1.5 TB SATA, 32 MB Cache, 7200 rpm
System disk	Samsung Spinpoint F1 1 TB SATA, 32 MB Cache, 7200 rpm
Optical drive	SATA DVD RW
6 GB Memory	3 × 2 GB 1600 MHz DDR3 Corsair TR3X6G1600C7D
Graphics card	Matrox Millenium G450 16 MB PCI
Motherboard	Gigabyte GA-EX58-UD5, ATX, Intel X58 chipset
CPU	Intel Core i7 920 S1366, 2.66 GHz quad core, 8 MB cache
Extended ATX Case	Thermaltake XASER VI
2 Cooled HDD enclosures	IcyBox Backplane System for 5 × 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

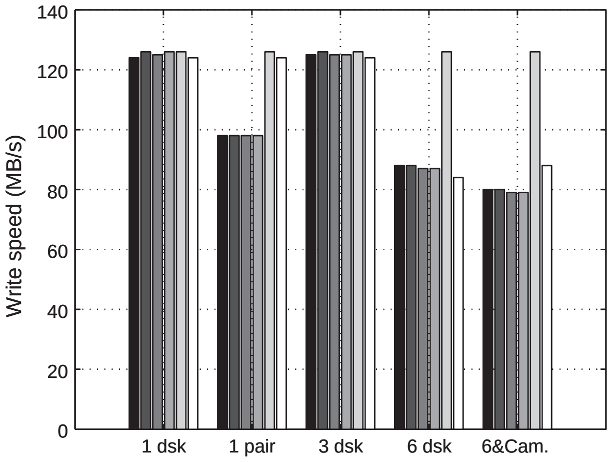

Fig. 7. Sequential disk write performance of the 6 SATA ports on the motherboard (ICH9R controller) for different configurations. From left to right, the WTR of ports 1 to 6 are shown, shaded from black to white. '1 dsk': writing to 1 HDD at a time. '1 pair': writing to 2 disks simultaneously, through 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 780 × 580 pixels at 60 fps.

was used for each HDD, set to write with data blocks of 256kB. Fig. 7 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 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. 7), although the 5th SATA port is able to maintain the full 124 MB/s of the HDD under all of the tested circumstances. This might be related to the fact that the ICH9R has two SATA controllers of which one supports four disks while the other supports two [28]. Connecting 3 disks to SATA ports of different pairs ('3 dsk' in Fig. 7) also allows us to write at full HDD speed. When writing to all 6 HDDs, while simultaneously receiving video from 18 cameras at 60 fps ('6&Cam.' in Fig. 7), the minimum WTR to each separate disk was 79.3 MB/s. This means that the system could store up to 475 MB/s of data, with all disks writing at the same rate.

4.3.2. A/V capture throughput results

The maximum throughput of 475 MB/s, found above, only holds for sequential writing from a single source in 256 kB 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 40.1 fps (313 MB/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 49.9 fps (346 MB/s), and with 14 cameras we could reach the full camera frame rate of 61.7 fps (375 MB/s). The CPU load during these tests was around 70%.

When streaming the data from 3 cameras at 61.7 fps (26.6 MB/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 Fig. 5). This means that, with 14 cameras at full speed, the usable storage size is only 571 GB 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 7.6 h at 61.7 fps.

4.3.3. 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 1024 × 1024 pixels and 59.1 fps, we made a few modifications, shown in Table 5. 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 8 extra SATA HDDs. Sequential WTR of this HDD controller was found to have a limit of 840 MB/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 830 MB/s for a maximum recording duration of 6.7 h.

It is important to note here some issues that we encountered while testing this motherboard. First of all, the Fujitsu RAID controller we used in the experiments has the product code S26361-F3257-L8. However, a similar controller with code S26361-F3554-L8 did not work together with the GA-EX58-UD5 (using their firmware available in May 2011). Secondly, under certain conditions, we encountered muted periods in the audio recorded either with the MOTU board connected to a FireWire bus or with the internal audio device of the motherboard. This happened when either enabling the processor's hyperthreading functionality, or when using a Stingray camera connected to a separate FireWire bus. It was unrelated to the amount of captured data. These experiences demonstrate the importance of compatibility testing between components.

5. Sensor-level synchronisation

To synchronise the data captured from multiple sensors, it is crucial to be able to relate all data capture times with each other. However, when different sensors are capturing with separate clock sources, a method is required to relate the different clocks. This section describes how to solve this problem for sensor hardware which either provides its capture trigger as an electrical output signal, or is able to measure an external sensor trigger signal synchronously with its sensor data.

We start with a formal definition of clock synchronisation in Section 5.1, followed by a description of our practical implementation of A/V synchronisation in Section 5.2. Section 5.3 presents the experiments and results of our A/V synchronisation at sensor-level, and we end the section with a short discussion in Section 5.4.

5.1. Definition of clock-synchronisation

Let t_i be the time according to a clock i and t_j the same time according to another clock j. The clock time t_j is related to the clock time t_i through a clock-mapping function, which can be approximated by an n-th order polynomial f_{ij}: $\mathbb{R} \rightarrow \mathbb{R}$:

$$t_j \mapsto a_{ij}(0) + a_{ij}(1)t_i + a_{ij}(2)t_i^2 + \ldots + a_{ij}(n)t_i^n \qquad (1)$$

where $a_{ij}(n)$ is the n-th order polynomial factor for the mapping from clock i to clock j. Considering the common use of highly regular crystal oscillators in hardware clock generation [29], it is reasonable to assume that, in practice, polynomial factors greater than $n = 1$ can be neglected. In that case, $a_{ij}(0)$ corresponds to a constant clock offset, and $a_{ij}(1)$ to a constant clock ratio. This means that synchronisation between clock i and clock j comes down to finding the relative clock offset $a_{ij}(0)$ and the clock ratio $a_{ij}(1)$. This can be solved with a minimum of two time-correspondences between the clocks.

5.2. A/V synchronisation in our setup

To synchronise between sensors, we centrally monitor the timings of all sensors, using the MOTU 8Pre external audio interface [30], connected to the capture PC through an IEEE 1394a connection. Since the analogue inputs of the 8Pre are sampled using hardware-synchronised

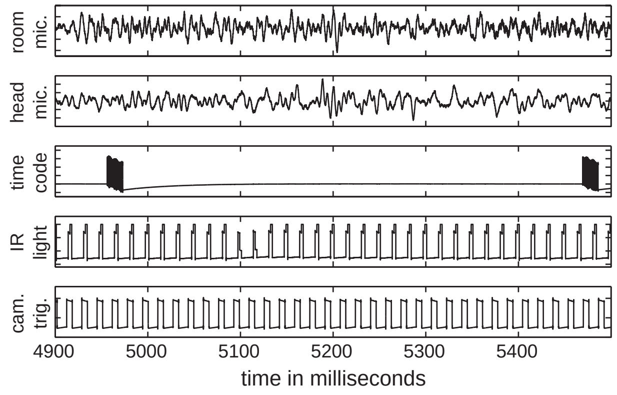

Fig. 8. 5 tracks recorded in parallel by MOTU 8pre audio interface. From top to bottom: (1) room microphone; (2) head microphone; (3) serial port timestamp output (transmitted at 9600 bps), showing 2 timestamp signals; (4) measured infrared light in front of eye tracker; (5) camera trigger.

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, 96 kHz. For our application, we used a sampling rate of 48 kHz. This provides a 20 μs granularity in determining signal timing.

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 Fig. 8) can be located in the audio signal with an accuracy of 1 audio sample. This means that, with an audio sampling rate of 48 kHz, the uncertainty of localising the rising camera trigger edge is around 20 μs. The frame exposures of the slave cameras start around 30 μs later than the triggering camera, with a jitter of 1.3 μs [22]. When this is taken into account, the resulting synchronisation error between audio and video can be kept below 25 μs.

An even more accurate A/V synchronisation is achieved by using all the detections of the regular camera trigger pulses over an entire recording to compute the clock offset and clock ratio between the audio sampling and video sampling.

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. Combining these two timing sources guarantees accuracy and robustness: Recording the sensor trigger signal in a parallel audio channel provides temporal alignment of sensor measurements with microsecond accuracy, while a sample's id or recording timestamp will allow us to detect, identify and account for any samples that might have been lost during transmission or data capture.

5.3. Results of sensor-level synchronisation

In Section 2, we have compared the synchronisation accuracy of our system to custom built research setups. However, as far as practical usability is concerned, none of the compared systems comes close. First of all, our system only requires COTS-hardware, plus a few straightforward cable connections. Secondly, most of the other systems are limited to video recording only. When it comes to recording a single camera view together with audio, there is no doubt that the most practical alternative to our system is using a commercial audio/video recording device,e.g. a camcorder, a webcam, or a photocamera with video recording capability. Such solutions are low-cost and require a minimal amount of effort to set up and use.

To compare the resulting synchronisation of commercial solutions to our own system, we have set up an experiment using a Shimpo 311A stroboscope. This produces light flashes with a duration of approximately 10 μs. This short flash duration will be captured in no more than one video frame that is captured by a camera. Furthermore, since the strobe flash is caused by ionising the gas in the xenon bulb, the sudden temperature increase also causes a pressure change that produces a clear 'tick' sound at the same time as the flash. The fixed delay of 0.61 ms between a flash and the detection of a 'tick' sound in the audio stream was determined from a comparison with the peak signals from a photo diode.

Setting the stroboscope to flash twice per second with a timing accuracy of 1×10^{-5}%, we recorded one-hour-long sequences with several different A/V recording devices. After transferring the data to a PC, each of the tested commercial A/V recorders delivers a recorded sequence of data as a single media file that contains both the audio and video samples. From the flash and tick locations in the beginning and end of the video and audio streams, we could accurately derive the actual video and audio sampling rates, respectively. Combining the detected moments of at least 50 strobe flashes in the beginning and the end of each one-hour recording, the actual sampling rates could be estimated with an accuracy of 0.004%.

In Table 6a, b and c, we compare the measured sampling rates with the rates that were reported in the headers of the media files. The Stingray camera we used does not report the sampling rate to which it was set. This is derived from the recorded trigger signal. Note that an

Table 6

Comparison of timing accuracy in commercial A/V recording equipment and our method of synchronisation by recording the camera frame exposure signal in one of the audio channels (5th column). A stroboscope was used to record audio-visual time markers. The measurements in a, b and c come from one recorded sequence of one hour. The statistics in d and e are estimated over 30 recordings of around 30 s each.

		Webcam	Photo camera	Camcorder	Frame exp. in audio
	Brand type	Philips SPC 900NC	Casio EX-F1	JVC GR-D23E	MOTU/AVT 8Pre/Stingray
a	Reported audio rate (Hz)	44,100	44,100	48,000	48,000
	Measured audio rate (Hz)	44,102	44,078	47,999	47,998
b	Reported video rate (fps)	30	29.97	25	Not required
	Measured video rate (fps)	30.00018	29.98619	24.99975	61.75259
c	Error in ratio of reported A/V rates	$3.69 \cdot 10^{-3}$%	0.10%	$5.51 \cdot 10^{-4}$%	Continuous-
	Multiplicative sync. error	133 ms/h	3.76 s/h	11.2 ms/h	Alignment
d	Average audio–video length	11.7 ms	18.9 ms	−19.8 ms	A/V recorded
	Standard deviation	25.4 ms	20.0 ms	5.25 μs	Independently
e	Average audio synchronisation lag	−17.27 ms	−11.6 ms	−37.5 ms	8.58 μs
	Standard deviation	19.3 ms	492 μs	416 μs	73.4 μs

Please cite this article as: J. Lichtenauer, et al., Cost-effective solution to synchronised audio-visual data capture using multiple sensors, Image Vis. Comput. (2011), doi:10.1016/j.imavis.2011.07.004

accurate synchronisation between audio and video does not depend on the absolute deviation from the reported sampling rates. What matters is the difference of the A/V ratio between the reported and the real sampling rates. Inaccuracy in this ratio results in a multiplicative synchronisation error that accumulates over time. Table 6c also shows the accumulation of the multiplicative synchronisation error per hour.

One way to prevent a multiplicative synchronisation error is to estimate the A/V sampling ratio from the ratio between the recorded number of video frames and audio samples. This assumes that the audio and video streams correspond to exactly the same duration of recording. However, as Table 6d shows, this is not always the case. For 30 recordings of around 30 s each, we compared the recorded audio durations to the recorded video lengths, by dividing the number of video frames and audio samples by the actual A/V sampling rates we have estimated earlier. These results show that the lengths of the streams can only be used to prevent a multiplicative synchronisation error in case of the camcorder. And only if the constant difference of 19.8 ms is taken into account. A variability of the relative audio stream lengths makes this strategy ineffective for the webcam and the photo camera.

To estimate the additive synchronisation error between audio and video, we compared the frame numbers of video frames that contain a flash, to the locations of the ticks in the recorded audio stream. For this, we assumed that the beginning of the audio stream should start at the same time as when the first video frame begins to show when it would be played back to a human observer. This means that the audio ideally starts at half a video frame period before the middle of the capture time of the first video frame. The measured additive synchronisation errors in Table 6e reflect how the actual A/V synchronisation deviates relative to this assumed audio starting time. For all devices except the webcam, the estimated standard deviations of the additive synchronisation errors are in the order of the accuracy of our measurement method. This means that the measured standard deviation can only provide an upper bound for the actual standard deviation of the additive synchronisation error.

The two main sources of a possible A/V synchronisation offset in our setup are: 1) the difference between the frame exposure output signal of the camera and the actual camera sensor integration time, and 2) the synchronisation between the two channels of the MOTU audio interface that record the microphone sound and the camera trigger signal, respectively. If we can assume that these factors are constant over different recordings, the average audio synchronisation lag of 8.58 μs (see Table 6e), can be regarded as an upper bound on the additive A/V synchronisation error in our system.

5.4. Discussion on sensor-level synchronisation

Our results show that the additive synchronisation error of our approach, using separate COTS sensors, is over a thousand times smaller than any of the compared commercial A/V synchronisation solutions. Since our method always synchronises the data streams using correspondences over the entire length of the recording, there is no accumulation of a multiplicative synchronisation error. With other A/V recording devices that have a sufficiently constant additive and multiplicative synchronisation error, a comparably high accuracy of synchronisation could possibly be achieved through measuring the actual audio and video sampling rates as well as the A/V offset. However, one cannot assume from 30 test samples that the synchronisation offset will never be shifted over the duration of one or more video frames. In fact, this has happened four times during a set of 30 sequences of 10 s, which we recorded earlier with the photo camera. Such unpredictable synchronisation effects make it impossible to guarantee accurate synchronisation. Instead, our approach combines the low cost and ease of use of commercial sensors, with a reliable synchronisation accuracy that is suitable for low-level sensor fusion.

6. Computer-level synchronisation

For sensors that do not have a trigger output or signal input, such as the Tobii X120 Eye Tracker, the synchronisation method described in Section 5 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 [31]. 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 binary-coded timestamp messages that contain the momentary CPU cycle count time, and transmits them through the serial port. These timestamp messages are recorded in a separate audio track, in parallel to the microphone and camera trigger signals (see Section 5.2). Two examples of a timestamp message are shown in the third audio track in Fig. 8.

From two or more timestamp messages, the linear mapping can be determined that relates any temporal location in the audio recording to the time used for the timestamps of the sensorial data captured by the remote system. This allows the temporal location in all parallel recorded audio channels (e.g. sound, camera trigger pulses or timestamp signals from another PC) to be related to the time of the remote sensorial data capture system. To achieve this, the temporal start location of each timestamp message in the audio recording is paired together with the time of the remote system retrieved by decoding the message in the timestamp signal.

In Section 6.1 we will 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 in Section 6.2. Then, Section 6.3 describes 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. Section 6.4 presents the experiments and results for using this computer-level synchronisation to synchronise the Tobii eye gaze data with the audio data. Finally, we end this section with a discussion on computer-level data synchronisation in Section 6.5.

6.1. Serial port timestamp signal generation

A standard serial port (RS-232 compatible interface) is used to generate a timestamp signal every 0.5 s, at a bit rate that can be easily read with the utilised audio interface. In our recordings, we used the MOTU8pre at 48 kHz 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 exact frequency of bits on the output of the serial port differs slightly from the specified bit rate.

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:

$$\lambda_1 = T_w + N_1 / R + T_r \tag{2}$$

$$\lambda_2 = T_w + N_2 / R + T_r \tag{3}$$

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:

$$R = \frac{N_2 - N_1}{\lambda_2 - \lambda_1}$$

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

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 30 µs. However, occasional outliers from this average can occur when transmission is interrupted by another system event. The largest outlier we came across during 7 h of recording was around 25 ms. These occasional outliers can be discarded easily by using robust statistics with a large set of subsequent time-stamps to estimate the linear time mapping.

6.2. 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 Fig. 9). This is compensated by interpolating the 'off' level according to the steady-state level before and after the timestamp signal.

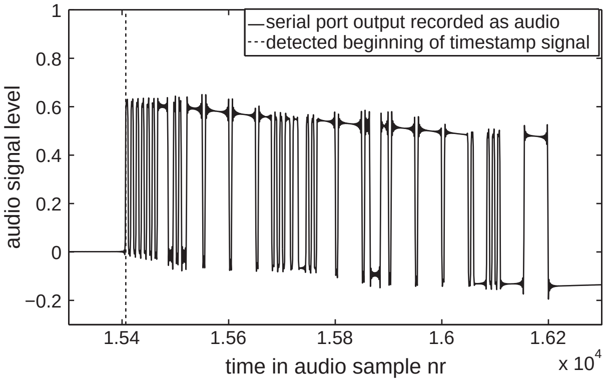

Fig. 9. A binary (on/off) timestamp signal output from a serial port, recorded as an audio channel. A high-pass filter used in the audio processing causes vertical skew along the timestamp, while an anti-aliasing filter causes ripple around the step edges. The skew is compensated before reading the timestamp message, by a linear interpolation of the steady state level before and after the timestamp.

6.3. 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 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, Fig. 10 shows the distribution of the time-difference of each individual timestamp compared to the linear regression on all timestamps in one of our recordings. This shows that most of the timestamp signals were received and correctly localised within 1 audio sample (20 µs) from the linear fit. Table 7a 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 Section 6.1, is unbiased, these results imply that an external system can be synchronised with an accuracy of approximately 20 µs. The actual accuracy will depend on the linear regression method that is

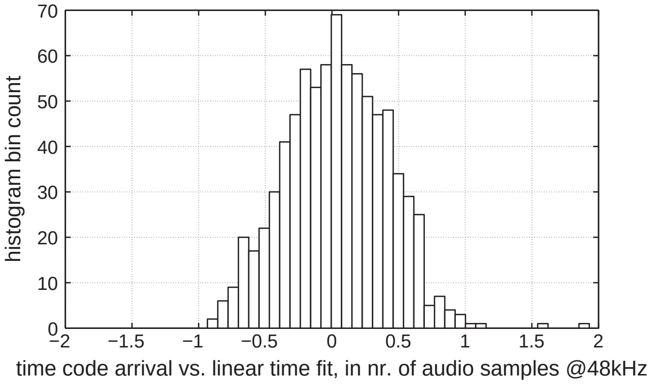

Fig. 10. Histogram of the time differences between detected onsets of timestamp signals from a linear fit to all timestamps in one recording. A timestamp signal is based on the CPU cycle counter of an external PC, transmitted through its serial port and recorded as an audio signal at a sampling rate of 48 kHz.

Please cite this article as: J. Lichtenauer, et al., Cost-effective solution to synchronised audio-visual data capture using multiple sensors, Image Vis. Comput. (2011), doi:10.1016/j.imavis.2011.07.004

Table 7
Statistics of estimated root mean square errors (RMSE) in 87 recordings of approximately 5 min long, measured in number of Audio Samples (AS) at 48 kHz 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/7.25 µs	0.546 µs	6.51 µs	9.38 µs
(b)	IR pulse time vs. its linearisation	0.235AS/4.90 µs	1.03 µs	3.44 µs	9.74 µs
(c)	Gaze data timestamp vs. IR pulse time	22.4AS/467 µs	287 µs	122 µs	1443 µs
(d)	Linearised gaze data vs. IR pulse time	15.2AS/317 µs	298 µs	35.9 µs	1412 µs

applied and on the length of a recording (the number of timestamp signals received).

6.4. Results for computer-level synchronisation

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 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 ('g' in Fig. 1), 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.

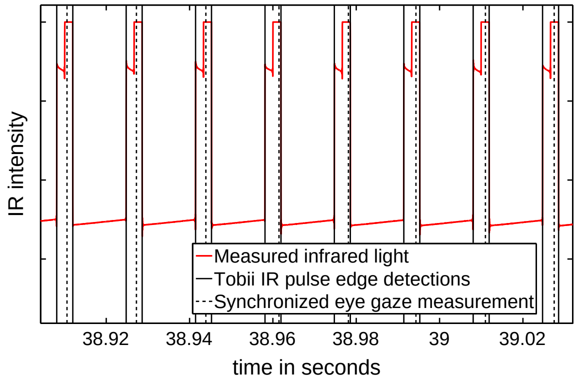

Fig. 11. Comparison between infrared light flashes from the Tobii X120 eye tracker (set to 60 Hz) and the timestamps of the recorded gaze data. The infrared light, measured by a photo diode in front of the Tobii X120, is recorded by an audio interface at 48 kHz. In this fragment, the largest deviation between the centre of the time interval of the IR flash and its corresponding data timestamp is 1.46 ms.

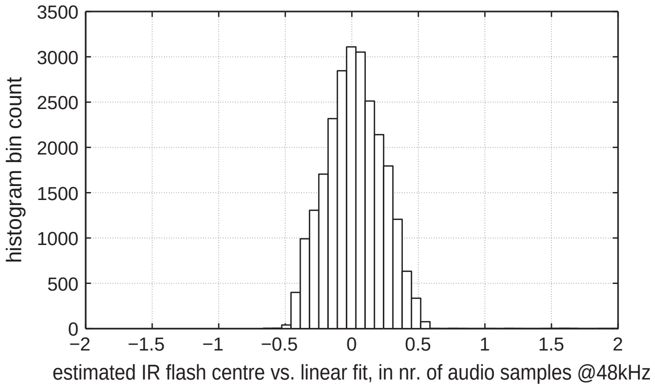

Fig. 12. Histogram of the difference of the estimated time interval centres of IR flashes, compared to their linear fit. Flashes were recorded with a photo diode connected to an audio input and placed in front of the Tobii X120 eye tracker.

An example of the comparison between the timestamps of the captured gaze data and the IR flashes is shown in Fig. 11. Note that, for our experiments, we have set the Tobii X120 to 60 Hz rather than 120 Hz, because this allows more freedom of head movement [31]. 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. Fig. 12 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 (10 µs at 48 kHz) from the linear fit. Table 7b shows statistics of the Root Mean Square Error (RMSE) measure over 87 recordings. The largest of all 87 RMSE estimates was 5.96 µs. 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 Section 6.3. Fig. 13 shows the progression of the estimated gaze data timestamp error 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

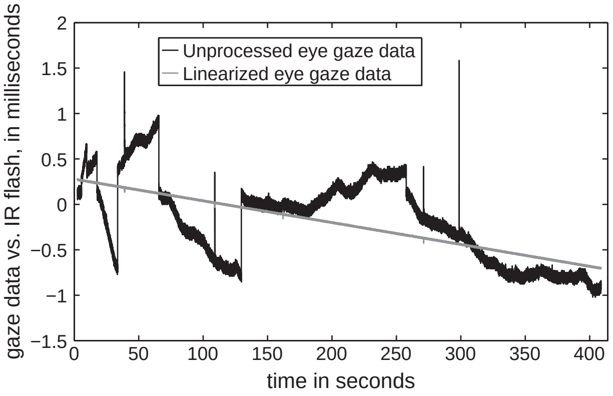

Fig. 13. Estimated gaze data timestamp error over time, in comparison to the interval centre of the closest IR flash, measured in the audio recording.

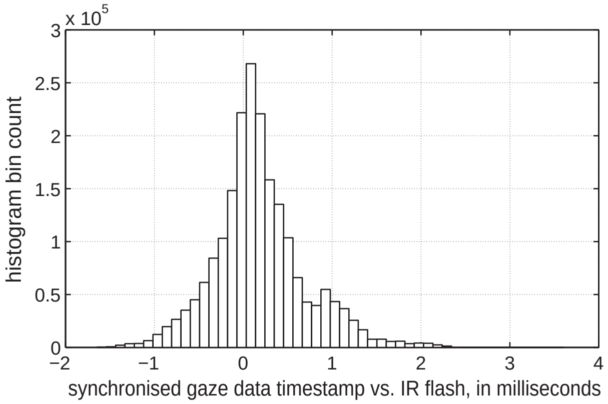

Fig. 14. Histogram of the difference between a gaze data timestamp and the time interval centre of the corresponding IR flash, measured over 87 recordings of approximately 5 min long. The most extreme offset measured among these 2,028,526 data samples was 3.60 ms.

before the incoming data is processed by Tobii Studio. Table 7c shows statistics of the RMSE over 87 recordings and Fig. 14 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 ms. 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 (10 μs at 48 kHz).

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 grey line in Fig. 13. The related statistics for the RMSE are shown in Table 7d. 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.

6.5. Discussion on computer-level 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 20 μs. 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.

7. 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.

Using low-cost COTS components, we built an audio/video capture PC that was capable of capturing 7.6 h of video simultaneously from 12 cameras with resolutions of 780 × 580 pixels each, at 61.7 fps, together with 8 channels of 24-bit audio at 96 kHz sampling rate. When capturing from 18 cameras, a bottleneck in the southbridge chip of the system's motherboard limited the frame rate to 40.1 fps. 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 1024 × 1024pixels at 59.1 fps, for a duration of 6.7 h. The captured data rate of this configuration amounts to a total of 830 MB/s.

For sensor synchronisation, we have proposed to use a multi-channel audio interface to record audio alongside the trigger signals of externally triggered sensors. Experiments showed that this approach can achieve a synchronisation error below 10 μs, compared to 11 ms or more for the compared commercial audio-visual recording solutions. For sensors without an external trigger signal, we have presented a method to generate timestamp signals with a serial port, allowing us to synchronise a PC that captures sensor data. Experiments show that the resulting synchronisation of a CPU cycle counter is accurate within 20 μs. 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 ms. Because the data was recorded at 60 Hz (with 16 ms 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 10 μs accurate.

Acknowledgement

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 27 (12) (2009) 1743–1759.
[2] H. Gunes, M. Pantic, Automatic, dimensional and continuous emotion recognition, International Journal of Synthetic Emotion 1 (1) (2010) 68–99.
[3] K.W. Grant, V. van Wassenhove, D. Poeppel, Discrimination of auditory-visual synchrony, 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, The Eleventh ACM international Conference on Multimedia, ACM, Berkeley, CA, USA, 2003, pp. 263–266.
[6] http://www.dvrsystems.net, website.
[7] http://www.boulderimaging.com, website.
[8] http://www.cepoint.com, website.
[9] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, ACM SIGGRAPH, Los Angeles, CA, USA, 2004, pp. 600–608.
[10] 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 Transactions Graphics 24 (3) (2005) 765–776.

Please cite this article as: J. Lichtenauer, et al., Cost-effective solution to synchronised audio-visual data capture using multiple sensors, Image Vis. Comput. (2011), doi:10.1016/j.imavis.2011.07.004

[11] 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) 319–323.

[12] T. Svoboda, H. Hug, L. Van Gool, Viroom — low cost synchronized multicamera system and its self-calibration, 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.

[13] 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 Processing (2009) 15. Article ID 351452, doi:10.1155/2009/351452.

[14] 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.

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

[16] 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, Journal of Physics D: Applied Physics 38 (2005) 2543–2555.

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

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

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

[20] H.-J. Freund, H. Büdingen, 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.

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

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

[23] S. Sookman, Choosing a camera interface: qualify and quantify, Advanced Imaging 22 (5) (2007) 20–24.

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

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

[26] M. Soleymani, J. Lichtenauer, T. Pun, M. Pantic, A multi-modal affective database for affect recognition and implicit tagging, Submitted to, IEEE Trans. on Affective Computing, Special Issue on Naturalistic Affect Resources for System Building and Evaluation, 2010 Nov. 8th.

[27] Anonymous, Gigabyte boards and DPC latency, Anandtech Forum, http://forums.anandtech.com/messageview.aspx?catid=29&threadid=2182171April 2008.

[28] Intel, Intel I/O Controller Hub 9 (ICH9) Family; Datasheet; Document Number 316972-004, August 2008.

[29] H. Zhou, C. Nicholls, T. Kunz, H. Schwartz, Frequency accuracy & stability dependencies of crystal oscillators, Technical Report SCE-08-12, Carleton University, Systems and Computer Engineering, Ottawa, Ont., Canada, November 2008.

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

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

Please cite this article as: J. Lichtenauer, et al., Cost-effective solution to synchronised audio-visual data capture using multiple sensors, Image Vis. Comput. (2011), doi:10.1016/j.imavis.2011.07.004