System for Artificial Tutoring of Songbirds

by

Leon Bonde Larsen

Submitted to
The Maersk McKinney Moller Institute
in partial fulfillment of
the requirements for the degree

Master of Science
in Robot Systems

at the University of Southern Denmark

Supervisors:
Associate Professor Coen Elemans
Associate Professor Poramate Manoonpong
Professor John Hallam

June 2016

Abstract

The zebra finch is a favoured model for investigating vocalisation. Especially the tutoring process where the young bird learns to copy the song of his father has been studied intensively. Many ingeniously designed experiments have been conducted in order to manipulate the stimuli presented to the bird during learning, but there are limits to what can actually be manipulated. Therefore scientists have been dreaming about a robot tutor that could give them control over more variables.

Building a vivid robotic version of a 10-30 gram bird is at least for the time being impossible. Several groups are working with simplified inanimate tutor robots and are getting interesting results, but development is slow and expensive.

Since the desire is really to control the stimuli, we instead chose another approach and built a teleconference system for birds. The two birds are physically isolated, but can communicate via cameras, screens, microphones and speakers.

In this work a communication system for birds mediating visual and auditory stimuli is developed. The system is intended for manipulative experiments with zebra finches to investigate the tutoring process. The system is developed and validated based on thorough analysis of both biological and technical issues.

Preface

In this thesis I present a robotics project. The project is interdisciplinary containing elements from both biology and engineering. It was conducted at University of Southern Denmark (SDU) under the Centre for BioRobotics, a collaboration between the Department of Biology and the Maersk McKinney Moller Institute.

The Danish educational system allows students to begin a PhD project before finishing the Masters programme; the so called 4+4 scheme. This way it is possible to do the Master project in synergy the PhD project.

The Faculty of Engineering at SDU has recently decided to implement the 4+4 arrangement so that the student conducts a stand-alone master project, reported in a normal Master Thesis. Under the old rules, the masters exam was conducted as a mid-term evaluation of the PhD project so in practice the two projects were conducted as one. The rules were changed immediately before I began my project and thus I am among the first students to do it the new way.

The new rules pose a risk of self plagiarism, since the PhD Thesis cannot contain material directly from the Master Thesis as under the old rules. In order to accommodate this risk, I have chosen to clearly separate the two projects. The Master Thesis exclusively concerns the technical development of the tools needed in the research, while the PhD exclusively concerns the research conducted using the tools.

The Master project constitutes a normal 30 ECTS Master of Science in Robotics, although I have been working on the system over three semesters as opposed to normally just one. Details and learning goals for the master project can be found at: `http://fagbesk.sam.sdu.dk/study/fagbasen/fagprint.shtml?fag_id=32225`

Being an interdisciplinary study, it has been important for me that this work can be understood by engineers as well as biologists. This inevitably means that some parts will seem obvious to engineers, but complex to biologists and vice versa.

The thesis is structured as a development project in two parts. The first part introduces the problem and conducts a thorough analysis of the desired system. The findings from the analysis are compiled into a requirements specification that would normally form a basis for the contract between costumer and manufacturer. Everything up to this point concerns work done before deciding to actually build the system. The second part of the thesis concerns designing and building the system as well as validating it based on the requirements specification.

I would like to thank my supervisors for daily inspiration, my colleagues for competent feedback and my family for their invaluable loving support.

Contents

Chapter 1

Introduction

Model organisms comprise very important tools in biological research. The most widely known lab animals are probably rats and mice, but different hypotheses require different animals. When studying the human brain, our closest relatives, the primates are often used as a model, but although similarities are great, there are important differences. One is that where humans learn to talk, the primates exclusively utter innate sounds.

In the study of vocal learning, a lot of behavioural and developmental testing can be done without harm on human subjects. However when investigating the brain at a functional level, a manipulative approach is often required and for ethical reasons cannot be performed on humanss. Important discoveries can be made based on individuals having suffered very specific injuries like strokes or brain damage, but can be very hard to generalise in order to draw conclusions.

In order to perform manipulative brain experiments, biologists have instead turned to our closest relatives with whom we share the ability to learn vocalisation. This group includes diverse species like dolphins, whales, elephants, bats and of course birds. Well known examples are parrots, mockingbirds and myna, but also less obvious species like nightingales, hummingbirds and songbirds. Many of these have been used as models in the investigation of different hypotheses.

The zebra finch (*Taeniopygia guttata*) is a subspecies of the songbird family and has proved to be an especially useful model (figure 1.1). Zebra finches are very easy to breed and they produce one very stereotyped song. In comparison, a nightingale typically knows more than a hundred different songs and thus quantifying learning can be quite difficult. In captivity the zebra finch father will usually tutor a juvenile male into exactly copying his song which makes it a good model for studying vocal learning. This tutoring process as well as the song development and brain structure

(a) In the wild

(b) In captivity

Figure 1.1: Zebra finch (*Taeniopygia guttata*).

of zebra finches have been heavily studied over the past sixty years and currently more than a thousand labs around the world carry out experiments on zebra finches.

An interesting observation about the tutoring process is that it works much better if the birds are able to interact with each other as opposed to only hearing each other (Baptista and Petrinovich, 1986). This is scientifically interesting since it is unclear how this interaction affects learning. At the same time good tutor birds can be hard to come by and any means to improve the learning process would be valued.

Some attempts have been made, with varying results, to build a tutoring robot in order to carry out manipulative experiments with tutoring (figure 1.2). This is a hard task, since a robot bird will need many degrees of freedom to be vivid. Building speaker, microphone, controller, sensors and actuators into a small bird shape proves both mechanically and electrically very difficult.

Figure 1.2: A juvenile zebra finch perched next to a plastic model. (Lipkind et al., 2013)

Instead propose another way of solving this problem is proposed based on a video

link. The idea is to have the two birds in separate soundproof boxes, while a system of cameras, screens, microphones and speakers will create an environment where the birds perceive themselves to be in cages next to each other separated only by a glass window, although really being in separate isolator boxes.

In Ikebuchi and Okanoya (1999) a similar idea was tested and it was shown that male zebra finches and bengalese finches would sing directed song to a TFT screen showing video of a female conspecific. It has also been shown that zebra finches would work for the reward of silent video images played back on a screen (Adret, 1997). Finally there are plenty of observations of birds interacting with mirrors, video or images of other birds (Personal communications with Michiel Vellema, Constance Sharf and Iris Adam). This suggests that the birds might believe such a virtual environment.

There are several advantages to this approach. At first glance it lets us carry out a number of experiments to investigate which modalities or combinations of modalities are important for interaction. For example we could study the effect of pitch, loudness, colour intensity, brightness, contrast, blurred vision etc. For example experiments manipulating the pitch of the song can teach us about the sound productive system, but so far such experiments have been based on masking certain frequencies with white noise (Leonardo and Konishi, 1999, Tumer and Brainard, 2007, Andalman and Fee, 2009). With a robot tutor we could go even further by artificially altering the tutor's song, exchanging syllables, slowing the song etc.

A big issue in many behavioural experiments is the tedious work of annotation. The tutoring process takes 50-100 days and the birds are typically active for 10-12 hours a day, so each trial produces vast amounts of data. In a system where all interaction takes place through a video link, the data is already conditioned for applying feature extraction and machine learning to do automatic annotation. For example it would be interesting to measure the number of tutoring sessions per day and follow how it changes over the tutoring period.

At a later stage it would be interesting to replace or partially replace the tutor with an artificial system capable of reacting to the behaviour of the juvenile. For example we could replay a response from an earlier recording in order to investigate the effect of unnatural responses. The system can even to some extent meet the desire for an artificial tutor, since it is much easier to generate sound and images than to actuate a 5cm mechanical bird.

It is beyond the scope of this thesis to test if the birds actually believe the system as well as to perform any manipulative experiments. Instead focus will be on the technical development of the system by first analysing the problem in order to reach a requirements specification and next to implement and validate the system according to that.

Chapter 2

Problem analysis

To further define and approach the problem it is necessary to get an overview of the setting in which the Artificial Tutoring System (ATS) will be used and the requirements imposed on the system. It is important to keep in mind that the system must be applicable to both experiments where two birds interact and experiments where the tutor is replaced by an artificial system. The aim of this chapter is to reach an elaborate requirement specification to support the implementation of ATS.

2.1 General design

2.1.1 Desired functionality

The ATS will be applied in scientific research concerning zebra finch communication. The main idea is to perform manipulative experiments wherein the system will monitor, assist or take over the tutoring task. This implies that the system should operate at a timescale where reacting to the behaviour of the birds is possible. Putting a number on the maximum delay is not trivial, so it must be considered in each of the sub-systems. Although the system can be used for data collection and offline analysis, focus is on the on-line, real-time use case.

Furthermore the system must be modular to allow extensions necessary for the specific experiments. Such modules are expected to be developed as stand-alone services subscribing to data already in the system and publishing processed data to the system. This implies following a publish-subscribe pattern as well as making the modules independent of each other. This means they can only rely on the presence of data, not on the module producing it.

As far as possible open source, modular solutions should be chosen and given a choice within this category large, active communities should be preferred.

2.1.2 Price

Although price is not a primary objective, the cost should be held at a reasonable level. There is no hard limit to this, but generally lab procurements up to 5.000 dkr. are easy to decide on. Most labs working with zebra finches have already invested in expensive sound recording equipment and reusing such equipment as a part of the ATS is preferable.

2.1.3 Scalability

Scalability is important in any system focused at scientific research. It is impossible to predict which modules will be needed, but it is fair to assume that they will need to subscribe to data from the system, publish data to the system or both. This makes the connection of the nodes in the system important, since the medium must be scalable, modular and independent of specific data formats. It should be noted, that some nodes might be embedded systems and thus communication should be applicable to distributed systems.

2.1.4 Modularity

The system is intended for research and should therefore be optimised for change. This implies a modular design, meaning that the system is subdivided into modules that can be rearranged in different ways (figure 2.1). For example a camera could be seen as a module and encapsulated with processing and communication capabilities so the same module could be applied in many different situations. Modularity also applies to the design of such modules, since for example the communication part of all modules might be similar so instead of reinventing it several times, it can be turned into a module and be applied in other modules at a higher level of abstraction. In general the developed system should be as open and decoupled as possible to simplify integration and modification.

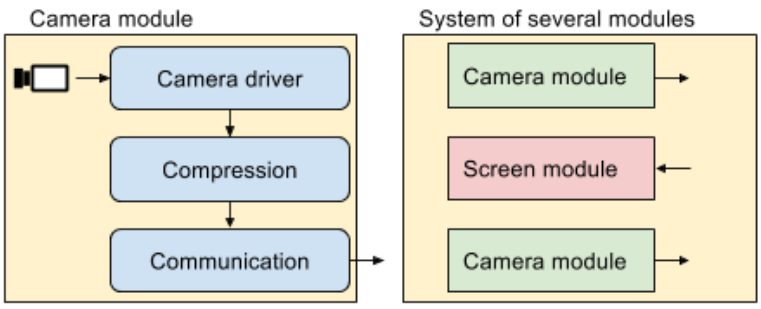


Figure 2.1: The modular design idea. Modules consist of sub-modules and systems are built from modules.

2.1.5 Accuracy

It is paramount in generating reproducable scientific data that the accuracy of measurements can be reported and to do so all parts of the system need to be properly calibrated and documented. With regards to the sound mediating system, the characteristics of both recording and playback equipment must be known as well as key values like delay and noise level. In the video part of the system it is important that colour and light intensity are properly reproduced and that the amount of delay is known.

2.2 The birds

2.2.1 Health and welfare

Zebra finches tend to peck, bite and eat any object presented to them; especially soft materials like glue, insulation or similar, but also uneven surfaces of wood or plastic. To counter this all surfaces must be smooth and properly coated.

The current legislation prescribes that procedures where animals are used in scientific research must be classified based on three levels of severity (directive 2010/63/EU). Behavioural trials like the ones expected to be carried out with this system however fall outside these categories and are thus in essence covered by the rules for husbandry. The main concern in behavioural experiments is isolation, since zebra finches are social animals. However given that they accept the proposed system, they cannot be considered socially isolated although physically isolated.

With regards to physical constraints OECD prescribes minimum $500cm^2$ per bird

(OECD, 2010) which should be observed in the lay-out of the cage.

2.2.2 Cleaning

During trials the bottom part of the cage is removed for cleaning twice a week. This is done by separating the cage from the bottom and sliding a piece of cardboard under the top part to keep the birds from escaping. Typically a spare bottom part will be used in order to move the top part directly to the clean bottom. This means that no part of the system can be placed on the floor of the cage and should only be fastened to the top part. Zebra finches generally tend to make a mess and thus delicate parts of the system must be properly protected from dust, faeces and water splashing. This essentially means following the International Protection marking code IP54 (IEC standard 60529).

2.2.3 Modalities

Zebra finches in nature live in trees where vision is often impaired and therefore the most important modality in their communication is sound. Vision has been shown to play a role in both choice of mate and in choice of tutor (Zann and Bamford, 1996). It is also apparent from the studies in Zann and Bamford (1996) that they use vision in order to recognise conspecifics and relatives.

There are important differences between the human visual system and that of the zebra finch. Whereas we have stereo vision and thus the ability to see in 3D, the zebra finch has its eyes on the side of the head leaving only a narrow field with stereo vision. However the zebra finch has an additional type of cone (figure 2.2) in the eye responding to light with a wavelength around 350-370nm (Bennett et al., 1996a). In comparison human vision is most sensitive to light around 550nm and we are completely blind to wavelengths below 400nm. It has been shown in Bennett et al. (1996a) that ultraviolet vision plays a role in mate choice, but it is unknown whether it influences communication.

Zebra finches have olfactory sense and its role in communication is also unknown, but the general belief is that they use it mostly to return to their nest (Caspers et al., 2013). The use of tactile senses is apparent from observations, since aggression is mediated either by pecking or pushing (own observations) and several studies report correlation between aggression and tutor choice (Slater et al., 1988). Neither olfactory nor tactile modalities will be mediated by this system. Instead the system will focus on mediating sound and vision being the most significant modalities.

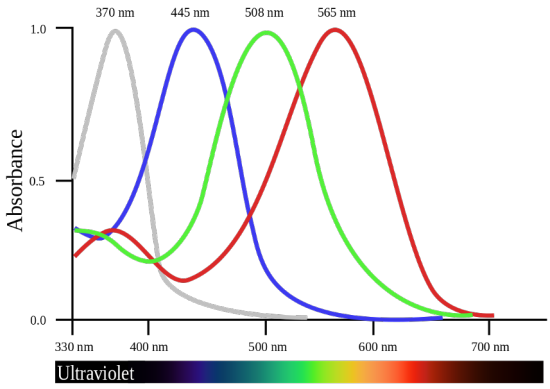
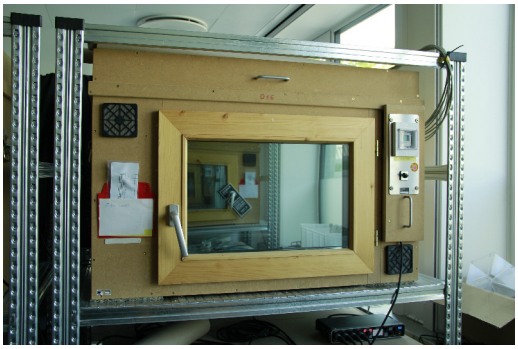


Figure 2.2: Shows the visual pigment sensitivity of birds with tetrachromatic vision, in this case estrildid finches (Hart et al., 2000).

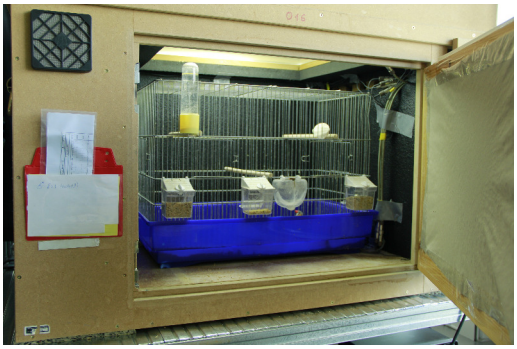
2.3 Auditory stimuli

2.3.1 Isolation

The birds need to be in acoustic and visual isolation during the trials and are therefore kept in isolator boxes like the one shown in figure 2.3.



(a) Closed



(b) Open

Figure 2.3: The isolator box. All interior surfaces are covered by sound absorbing material except the light source in the ceiling.

It is important to know the sound isolating properties of the isolator box in order to claim that the birds are actually communicating through the Artificial Tutoring System (ATS) and not just hearing sound from outside the box. This means that the audible frequencies must be damped according to the audible threshold for

zebra finches. Measuring the audible threshold is a very involved process either requiring training of the animal or measuring the auditory brain stem response of the anaesthetised animal while subjected to sound (Amin et al., 2007).

The audible threshold is reported as an audiogram like the one shown in figure 2.4, showing the barely noticeable sound intensity as a function of frequency. There are some variations between the available audiograms for zebra fiches, but a conservative conclusion is that attenuating frequencies from 500Hz to 8kHz by 40dB re 20 μPa will do. The isolation box should therefore dampen the sound by at least that amount.

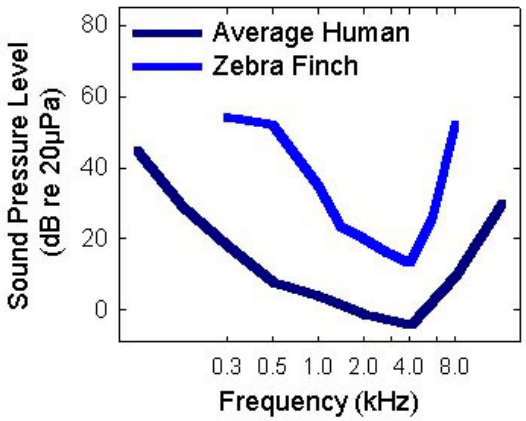


Figure 2.4: The zebra finch audiogram showing the barely noticeable sound intensity as a function of frequency (Laboratory of Comparative Psycoacoustics, University of Maryland).

2.3.2 Reverberation

Reverberations are caused by the sound being reflected off surfaces and thus the same sound arrives multiple times with different delays, at the same microphone. Reverberation is inevitable, but can be reduced by covering hard surfaces with sound absorbing materials. However the light source in the sealing of the isolator box cannot be covered and within the cage, the birds would peck and eat the material.

Reverberation can cause different problems. First the inherent feedback loop in the setup (sound from one box is played back and recorded in the other box to be played back in the first box again) might be triggered by reverberation. This is a common problem in live music sound engineering and is in general worsened with increased amplification, low frequency and reverberation (Troxel, 2005).

Next, reverberations in general diminish the quality of recordings and could make feature extraction harder. The reflections can to some extent be detected and removed at a later stage based on the auto-correlation of the signal, but at high sample rates this is computationally expensive and should be avoided when possible. In principle it would also be possible to decorrelate the signal with the impulse response of the box, but the response depends on the location of the sound source. However during interaction the bird is stationed in front of the screen so this might still be a viable solution.

2.3.3 Standing waves

Standing waves or sound nodes are caused by reverberations arriving at the microphone in opposite phase to the real signal. This ideally causes the two waves to cancel out, but in practice dampens part of the signal. The level of attenuation is a geometric property of the isolator box depending on reflecting surfaces, frequency and the position of the sound source and microphone. Standing waves will normally turn up in spectrograms as a missing frequency band. Signals degenerated by standing waves cannot be reconstructed. In normal recordings the nodes result in a lower accuracy in the measurements, even with well calibrated high-end equipment since it affects the measured source level. In recordings intended for automatic classification it might change the features enough to cause misclassifications.

2.4 Visual stimuli

2.4.1 Quality

The knowledge about the zebra finch visual sense in relation to video is very sparse, but we know a lot about their vision in general. From a mechanistic point of view birds have terachromatic or even pentachromatic vision (Bennett et al., 1996b) as opposed to humans trichromatic vision. In other words where the human eye has three classes of cones allowing us to see light of three different wavelengths (red, green and blue) the bird eye has at least another class making it sensible to wavelengths that are invisible to humans.

Zebra finches have their peak spectral sensitivity at 359nm (Aidala et al., 2012). Compared to humans that have peak spectral sensitivity ranging from 555nm in daylight to 505nm at night (Gross, 2005) the birds see light far into the ultraviolet

spectrum. This is a problem since cameras and screens lack an ultraviolet colour channel and generally are developed not to emit wavelengths below 400nm.

Hunt et al. (1997) investigated the influence of ultraviolet vision on mate choice and found that female zebra finches do not show preference to males with red headband when ultraviolet light is blocked. It has previously been shown that headband colour plays a role in the choice of tutor (Zann and Bamford, 1996), but the influence of ultraviolet vision on tutor choice and on tutoring in general is unknown. Since equipment for transmitting an ultraviolet channel along with the red, green and blue channels, would at best be extremely expensive and might even require new inventions, it is omitted in this setup. The assumption is, that given a choice the birds would prefer interaction with a tetrachromatic representation, but when unavailable, they will rely on other stimuli. This is a reasonable assumption given that both in nature and under lab conditions, the ultraviolet cues will sometimes be suppressed and therefore the species must be able to adapt.

Another difference is that birds have very large eyes relative to their head size. This means less room for muscles to move the eye and in most species including the zebra finch the eyes are fixed. Instead birds turn their head and along with the placement of the eyes, this means that only a small area in front of the bird has binocular vision. When birds are flying, they have been shown to rely on optic flow fields for depth perception (Eckmeier et al., 2008), but these cues are weak under static conditions like in the tutoring situation. In experiments where birds communicate through a window, they will often move close to the window and turn their head to look with just one eye (personal communication with Michiel Vellema). This suggest that the birds tend not to have good depth perception in static situations and thus presenting them with a two dimensional projection on a screen should be as live like as a three dimensional image.

Finally birds generally show higher flicker fusion frequencies (FFF). Humans have an FFF around 50Hz, meaning that a light source that is turned on and off more than fifty times a second will look like a constant light source to the human eye. Birds have been shown to have FFF from 60Hz up to 145Hz (Ikebuchi and Okanoya, 1999). Flicker fusion is exploited in many applications for humans including LED intensity control, screen technology and light tubes. It is well known in amateur bird husbandry, that some fluorescent lamps can stress the birds, presumably because they flicker at 100-120Hz. Ikebuchi and Okanoya (1999) investigated this by comparing male undirected song towards projections of female conspecifics. They found that using CRT screens (flickering at 60Hz) there was considerably less interaction that using TFT screens (not flickering). The solution should take this into account by using unflickering screens.

2.4.2 Camera view

Although not explicitly proven in zebra finches, several other bird species have been shown to rely on gaze estimation (Schmidt et al., 2011, Schloegl et al., 2007) and it is assumed that zebra finches are at least able to detect if the gaze is off (personal communication with Coen Elemans) making it more likely that they reject the illusion. The camera view must therefore be directly in front of the bird to properly mediate eye contact. This essentially means recording from a point behind the screen which might be complicated. This angle will also ensure the best possible footage of the communication.

2.5 Cage and isolator box

2.5.1 Physical dimensions

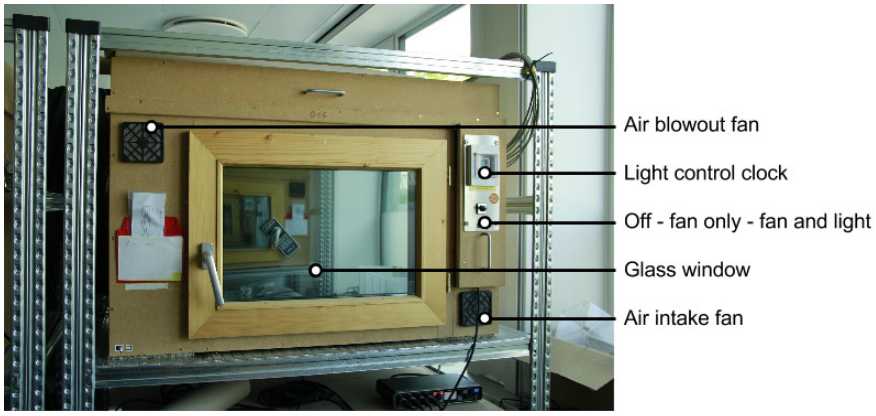
The cages and isolator boxes (figure 2.5) are available in the lab and have previously been used for acoustic experiments with zebra finches and canaries. The cages measure 30x40x60cm (26x34x53cm interior) and consist of two parts. The bottom part is a plastic tray and the top part is a metal bar cage. The isolator box measures 47x47x67cm (depth, height, width). It is covered internally on three sides with sound dampening foam, the roof is glass with a light source above, the door is glass and the floor is raw MDF wood. All wires to and from the inside of the box must pass through a Ø2.5cm hole, although it would be a small change to widen the hole.

The box is ventilated using low noise fans blowing air into an adjacent sound isolated chamber and from there into the main compartment. This is to further reduce the fan noise in the box and make them ideal for recordings.

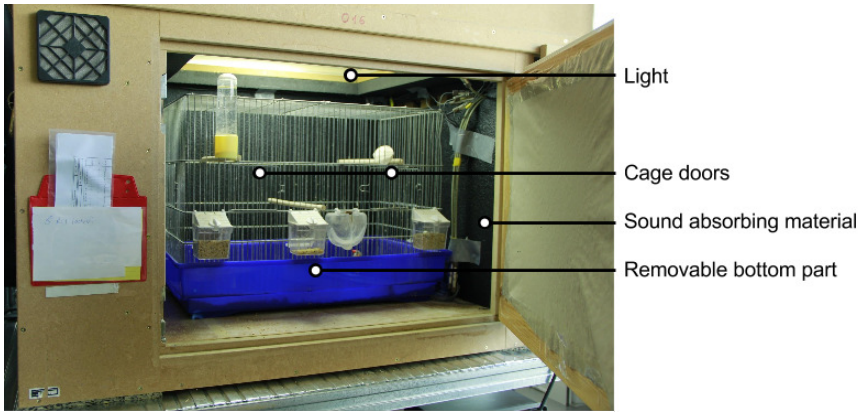
2.6 System architecture

2.6.1 Real time

The ATS is intended for manipulative behavioural experiments wherein it needs to interact with a bird. This is a big difference compared to the numerous studies based on off-line data analysis. It imposes the important constraint on all parts of the system that they must run close to real-time.



(a) Isolator



(b) Cage

Figure 2.5: Shows the different parts of the isolator box and the cage.

Although today's computers are quite fast and seem able to do many things at once, it is important to understand that this is merely a clever way in which the operating system shares resources among different tasks. In reality the computer still chews through all the calculations one at a time. The concurrency illusion is in practice created by the scheduler program that shifts processes in and out without them even noticing it. This is generally a problem for real-time systems, because a process needs to wait until it is given computing time. Some operating systems can handle real-time processes, but it tends to add complexity and overhead to the task. An alternative approach is to build a distributed system consisting of many small computers carrying out sub-tasks in true concurrence instead of one big computer handling everything.

A distributed system where cheap sub-modules can be easily replaced or reassembled into other systems and where individual parts can be developed and tested outside

of the system also adheres to the modularity design criterion. Scalability is also improved in a distributed system.

2.6.2 Synchronisation

When dealing with video and sound intended for humans, synchronisation is a big issue. There are different opinions on the limits for when humans notice the asynchrony. The Advanced Television Systems Committee recommends that 'audio should lead video by no more than 15ms and audio should lag video by no more than 45ms' (Television and Committee, 2003). The International Telecommunication Union found the threshold for detectability to be -125ms to +45ms for 'expert users' (ITU, 1998) while it was found in Younkin and Corriveau (2008) to be 185.19ms, with a standard deviation of 42.32ms for the 'average user'. The effect of synchronisation in zebra finches or even in birds is unknown so the delay should be kept at a minimum, but measured and reported.

2.6.3 Data

The data produced by the system will be of very different types ranging from continuous streams of video and sound to discontinuous extracted features or reasoned states, but there are also similarities. The system is generally dealing with data of a time-linear nature and should take that into account. It implies that instead of thinking about data as files with beginning, end and size, it should be thought of as continuous streams. Some streams will only be published when certain conditions are met and thus be discontinuous, but a time stamp can still associate such data with the continuous media stream. An example of such discontinuous data could be the subtitles for a movie. Formally data is finite and thus streams are in fact codata, but the following discussions will distinguish only between finite batch-data processed together and infinite streams processed as they become available.

Each node works on the current flood (consisting of one or several streams) either producing to it, consuming from it or both. If a node for some reason needs access to past data, it has to implement this itself and be present at the time of recording. No common means of time seeking will be implemented at this point.

In the general case, each stream should consist of only one type of data, since the consumer should be unaware of the producer and vice versa. However it could be that a consumer needed to receive synchronised data or it might be more important to quickly offload data from a producer in order to keep up with the real time

constraint. In such trade-off situations the real-time constraint is more important than the clean design.

2.6.4 Operating environment

Since most of the equipment will be mounted inside the cage with the birds, it is generally not accessible. This calls for robust systems that can run for hundreds of days with only remote maintenance. Over long periods, failing power supply could occur and the system must be able to recover without intervention once power is restored.

2.6.5 System control

The overall system control is responsible for initialising, connecting and monitoring the individual machines. The simplest approach would be to fix all addresses in the system and make each of the machines run a fixed script upon boot. Although this would work, development and maintenance of the system would be cumbersome and it would reduce scalability. Instead a solution with a temporary master is desired. The master need only be present upon monitoring or maintenance, but should be able to carry out those tasks remotely. It should be possible to work on scripts and programs off-line on the master and then push the changes to all parts of the system.

Chapter 3

Requirements specification

The following table summarises the requirements extracted from the analysis and supports the implementation. The system will be inspected and verified based on each point in the specification. The masure and type of inspection performed is noted in the table. The specification also forms the basis of a contract between the person ordering the system and the developer building it.

The deliverable is a communication system for otherwise birds mediating visual and auditory stimuli. The system is intended for manipulative experiments with zebra finches to investigate the tutoring process.

REQUIREMENT		MEASURE	INSP.
General design			
R-1-1 :	Modules independent	Can only depend on data	Design
R-1-2 :	Expandable	Follow publish-subscribe pattern	Design
R-1-3 :	Open source	All licenses must be open	Design
R-1-4 :	Reasonable cost	Less than 5.000 dkr per setup	Cost
R-1-5 :	Reuse existing recording equipment	No hardware dependency	Design
R-1-6 :	Distributed	Modules communicate via network	Design
R-1-7 :	Documented accuracy	All parts must be calibrated	Design
The birds			
R-2-1 :	Practical for cleaning	Smooth, cleanable surfaces	Design
R-2-2 :	Safe	No soft parts to peck	Design
R-2-3 :	Suitable size	At least 500cm^2 per bird	Design
R-2-4 :	Convenient husbandry	Bottom of cage removable	Design
R-2-5 :	Protected equipment	IP54	Design
Auditory stimuli			
R-3-1 :	Acoustic isolation	-40dB re 20μPa @ 0.5-8kHz	Test
R-3-2 :	Minimal reverberations	Unnoticeable to the human ear	Auditive
R-3-3 :	Homogeneous sound field	Variations less than 10μPa	Test
Visual stimuli			
R-4-1 :	Flicker-less screens	High speed camera	Test
R-4-2 :	Eye contact mediation	Correct camera placement	Design
Environment			
R-5-1 :	Fit in isolator box	Cables pass through Ø25mm	Design
R-5-2 :	Fit in cage	Max 26x34x25cm	Design
R-5-3 :	Low fan noise	Sound inaudible	Test
Architecture			
R-6-1 :	Low latency	Delay less than 150ms	Test
R-6-2 :	Synchronised stimuli	Smallest possible	Design
R-6-3 :	Support continuous data	All data as stream	Design
R-6-4 :	Long term deployment	Up for more than a week	Test
R-6-5 :	Robust to power loss	Can recover from power loss	Test

Chapter 4

Implementation

4.1 Architecture

4.1.1 General design

The system consists of two identical Artificial Tutoring Systems (ATSs), one for each end of the communication. The ATS consists of an Interaction Interface Box (IIB) installed in the cage (figure 4.1 b), a multi-channel recording array installed in the isolator and network connecting everything (Figure 4.2). Modules for monitoring, storing, processing or controlling can be added to the network, but are not discussed here.

(a) Both IIBs (b) Closeup on one of the IIBs

Figure 4.1: The two Interaction Interface Boxes with a toy bird looking into one of them.

This complies with R-1-6 being a distributed system. Each of the small units does only one thing, but does it well. This is one of the tenets founding Unix operating systems and

it supports the idea of modularity, since the small units can be connected in many different configurations. It also eases development and modification that modules are independent (R-1-1).

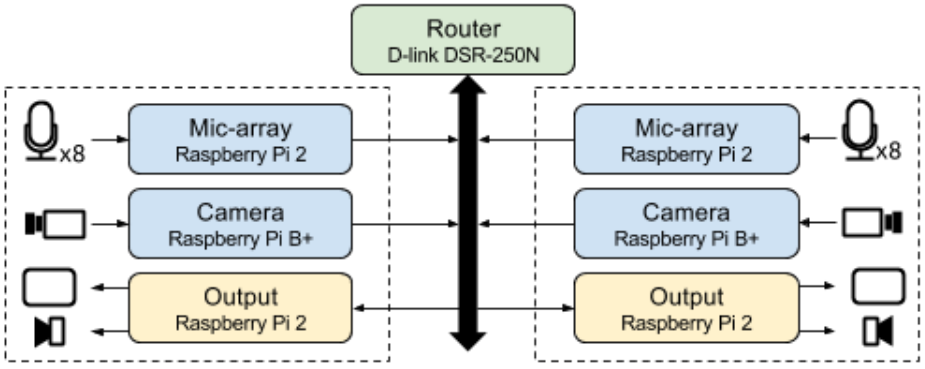


Figure 4.2: Shows the hardware architecture of the two ATSs, each consisting of eight microphones, video camera, display, speaker and three computers.

4.1.2 Design pattern

The system architecture follows the publish-subscribe messaging-pattern (Hasan et al., 2012, Birman and Joseph, 1987) meaning that the producer of messages is ignorant of the consumers and vice versa (R-1-2). Instead the producers publish data on one or more topics and the consumers subscribe to one or more topics. A module can be both a consumer and a producer -for example if it subscribes to data, performs data analysis and publishes the results. The advantage of this design pattern is a loosely coupled and very modular system, since no module depends on another specific module to function; only on the data (R-1-1). At the same time the system increases scalability since several modules can work independently on data from the same source. To further ensure low coupling and at the same time implement the continuous nature of the data, all topics are implemented as streams, meaning that they are associated through common time stamps (R-6-3).

In most implementations of the publish-subscribe pattern, there is a designated machine called the master orchestrating how the system is set up (Quigley et al., 2009). This master would for example handle name resolution of topics or machines in the system as well as directing service calls and monitoring. In other words the main reason to have a master is to exert control. It is important to notice that although the system whilst running is subject to near real time constraints, the system control task is not. Therefore control can be executed from a temporarily connected machine or even remotely. Another important task for the master is to handle the network topology. For example when a topic becomes

available, it is registered with the master and the master will notify the subscribers and provide connection details about the publisher. In a system with fixed topics, this can be avoided, if the publisher sends out one copy to the router and the router keeps a list of subscribers.

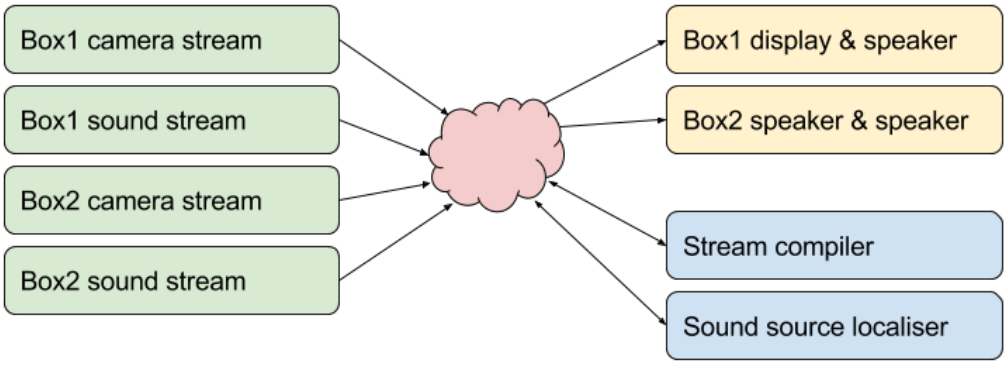


Figure 4.3: Shows the streaming architecture. Some units publish streams to the network, some subscribe to streams and some do both (double arrows).

4.1.3 Network

In normal Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) connections, the sender is responsible for copying the package and sending it to each of the receivers. This means the more consumers, the more workload and thus the system loses scalability (R-1-2). Another alternative would be to broadcast the data meaning that the copying is done in the router instead. This takes the load of the producer, but now the load is on the consumers instead, since everyone receives each packet. Although dropping an uninteresting packet is a very fast operation and thus will not significantly load the receiver, it would still lead to network congestion at the receivers interface since the uninteresting packets would steal bandwidth from the interesting packets.

In the Internet Protocol version 6 (IPv6) and in IPv4 via class D addresses and ICMP spoofing, another option called multicast is implemented. Multicast has the advantage of making the system truly scalable, since the distribution of data is done by the router, but is only sent to consumers that subscribe to it. In practice the consumer tells the router that it wants to be part of a certain group identified by a specific IP address. Whenever someone sends packets to this address, the router passes a copy of the packet to each of the registered receivers. In this approach, the concept of a topic is implemented as an IPv6 address. Keeping track of which IPv6 address contains what data is a task for the master, but in this relatively small setup the addresses are statically allocated and embedded into the applications.

The choice of transport layer protocol is between TCP and UDP. In order to understand the difference, it is necessary to understand the conceptual Open Systems Interconnection (OSI) model (figure 4.4) that forms the basis for networking (ISO/IEC 7498-1). The idea is that each layer at one end communicates with the corresponding layer at the other end. For outgoing messages a layer wraps the message received from the layer above and passes it on to the layer below. For incoming messages, a message received from the layer below is unwrapped and passed to the layer above.

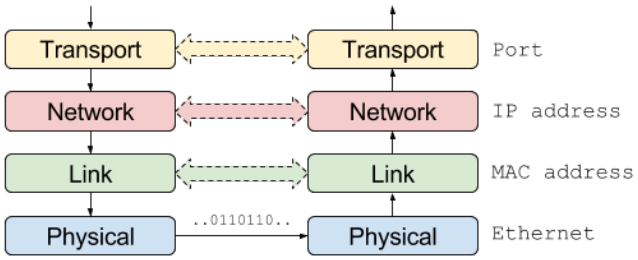


Figure 4.4: Shows the bottom four layers of the seven layer OSI-model.

The lowest layer is the physical connection or more specific the electric signal transmitted via cables or radio waves. Above that is the link layer connecting interfaces designated by MAC addresses. Next is the network layer connecting machines via IP addresses and above that the transport layer connecting processes designated by port numbers. If a packet is lost, it is the responsibility of the transport layer to detect it, typically based on sequence numbers and correct it for example by requiring the sender to retransmit.

Working on continuous data in the form of streams retransmitting data as done in TCP is unnecessary since the data will be obsolete by the time it arrives and it might even cause newer data to be delayed. Alternatively UDP does none of these and instead relies on sending small super light weight packets. If packets are lost, no one knows and no one cares. For completeness it should be mentioned that other protocols are available like the Reliable Data Protocol (RFC908) and Reliable User Datagram Protocol, but these are not commonly used and thus might cause incompatibility problems. For example the RDP protocol is not implemented in RTP and Gstreamer. UDP was chosen as transport layer protocol for streams in this project (R-6-1), but in critical cases like logging and control, TCP is used.

4.1.4 Streaming container

Associating data from different publishers is critical in a distributed system. This means some data about the data (meta-data) needs to be added. In order to do that in a complete, consistent and concise way, the streams are wrapped in a container. The purpose of the

container is to encapsulate an arbitrary number of streams and be able to merge any combination of the streams into one.

The OGG container was chosen for this project because it provides this functionality with a small overhead and it is fully stream oriented, meaning that the container can be written and read in one pass. This is important both in interleaving of streams and in real time data analysis. One particularly important property of the OGG container is that the meta-data is contained in the container instead of being built-in as in many other containers. This means that the designer can choose what meta-data is relevant and implement just those thus ensuring minimal overhead. Finally the OGG container is searchable, which is important when handling big data with sparse annotations.

4.2 Hardware

4.2.1 Screen

The screen used in the setup is a backlit 7-inch LCD 800480 capacitive touch screen with HDMI connector bought on e-bay for £40. The backlight is non-flickering and thus complies with requirement R-4-1.

4.2.2 Camera

The camera is a high-definition Pi Camera v2. The sensor is an 8 megapixel Sony IMX219 capable of full-HD (1080p) up to 30 fps and VGA up to 90 fps. The camera is connected to a Raspberry Pi through a 15 pin flat cable. The camera has a hyperfocal distance of 2m according to specification, meaning that objects from half that distance and out to infinity will be reasonably sharp. Objects closer than half the distance will be blurred in the image.

The distance from the camera to the perch is 22cm and therefore a lens is needed to correct the focus. Lenses are classified based on either focal length or refractive power and there is an inverse relationship between the two. The refractive power P, measured in dioptre (m^{-1}), needed to correct the focus can be found from the initial focal length f_{init} and the desired focal length f_{des}

$$\frac{1}{f_{init}} \cdot P = \frac{1}{f_{des}} \;\Rightarrow\; P = \frac{f_{init}}{f_{des}} \tag{4.1}$$

$$P = \frac{2.0}{0.22} = 9.09m^{-1} \;\Rightarrow\; f = 0.11m$$

This refractory number is merely a guideline, since there are many sources of errors. For instance the focus range of the camera is from two meters to infinity, but there are different definitions of this. The choice of lens should therefore be augmented by actual tests.

In order to have cheap lenses of different values, glass from normal reading spectacles was used. Under the assumption that there is no distance between the lenses, they can be stacked and thereby combined into different refractory powers. This is not exactly true, but it was found for practical purposes to be acceptable.

A test was conducted where a toy bird was placed at a distance of 22cm from the camera. This corresponds to the expected distance of the bird and using different lenses, still images could be visually compared. It was found that the image was sharpest with a refractory number of 9.0 dioptres combined from lenses of $2.5m^{-1} + 3.0m^{-1} + 3.5m^{-1}$.

4.2.3 Computers

A range of different small board computers were compared both on specifications and on maturity with respect to hardware and available software. It turns out that there are huge variations in the maturity and the community support, which in the end became the deciding factor.

The specs on the other hand turned out to be less important. For computing power and memory all tested SBCs were above 700MHz and 512MB RAM, which is sufficient to run the the required tasks. For the output board however, a multi-core SBC was desired in order to run two tasks (displaying video while playing sound). The ethernet and GPU thus became the most interesting specification, since there was a choice between faster networking in order to send uncompressed data or slower networking but increased demand on computing power for encoding and decoding. This narrowed it down to using either the Hummingboard with gigabit network or the Raspberry Pi 2 with only 100Mbit/s.

It turned out that the hummingboard had several issues running the software required for this project. The debian based Jessie distribution had issues with drivers for the SGTL5000 used for onboard sound on the Hummingboard. This worked in the earlier Wheezy distribution, but gstreamer depends on the newer kernel in Jessie. The Cubox distribution maintained by SolidRun (the company behind Hummingboard) could not be intstalled on the available older Hummingboards. All these problems could be solved by rebuilding kernels, porting drivers and a lot of debugging over several weeks.

In comparison installing the latest Jessie image on the Pi2, building openMax drivers for GPU based encoding and decoding and getting everything else to work took less than a day. Therefore the choice was on the better community and more mature software instead

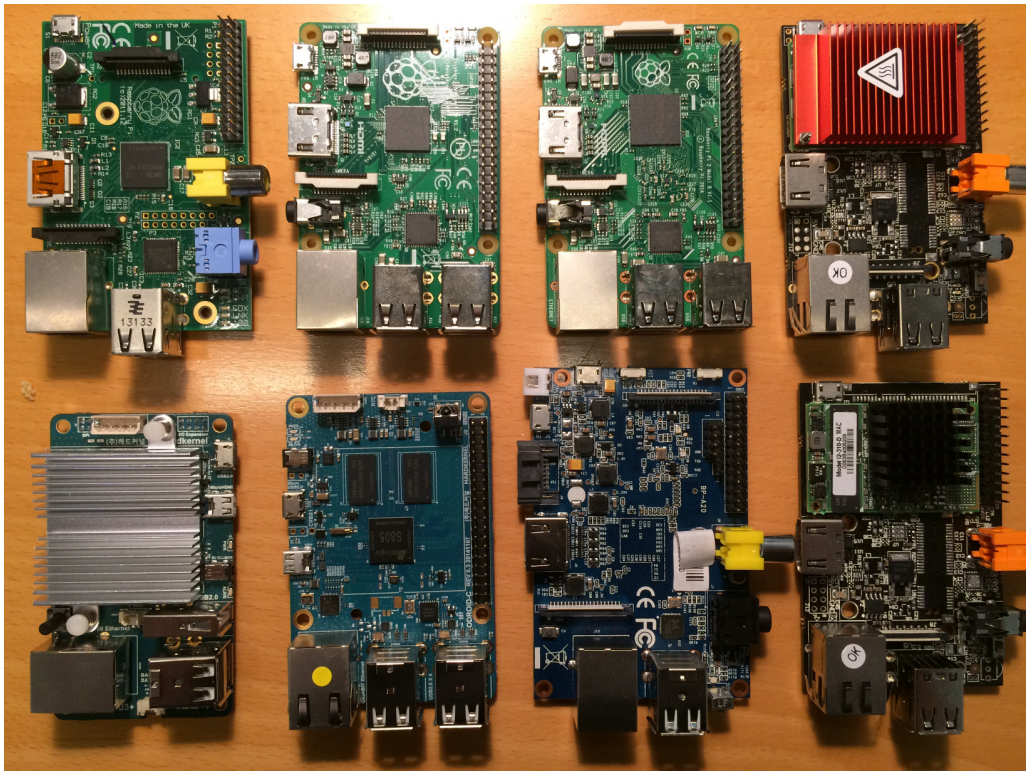


Figure 4.5: The evaluated SBCs. Top row from the left: Pi, Pi B+, Pi 2 and HummingBoard Pro. Bottom row from left: Odroid-U3+, Odroid-C, Banana Pi and Hummingboard Base.

of the better hardware. Thus each IIB ends up consisting of a Pi1 model B+ to stream data from the camera and a Pi2 model B to play back sound and video.

4.2.4 Network

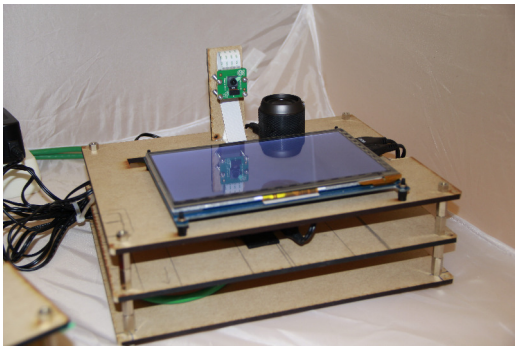
Since all data is passed via the network, this is an essential part of the system. The network consists of a router and potentially a number of switches connecting all nodes via IEEE 802.3 ethernet. The main thing to consider is the speed of the network. There are several speed classes, namely 10base-T, 100base-T, 1000base-T, 10Gbase-T and 40Gbase-T. The numbers refer the supported speed in Mbit/s or in the two latter cases in Gbit/s. The neat thing about ethernet is that it is backwards compatible, so a 100base-T interface can be connected to a 1000base-T router.

Apart from data transfer rates, the main implication of network speed is congestion. Network congestion is when the network is handling more packets than its capacity, which

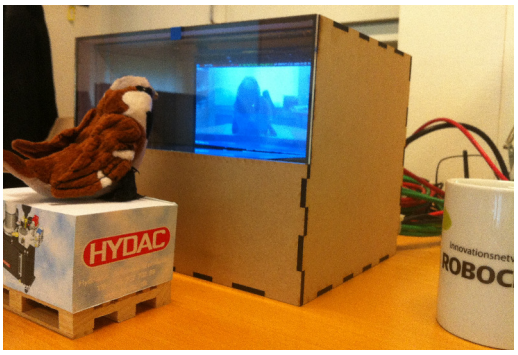
increases queueing latency and increases risk of data loss. It is possible to calculate the required network speed, but it requires explicit knowledge about the data, compression settings etc. Those details are unavailable in the present case and being a research project they would inevitably change anyway.

The chosen hardware is an eight port D-link DSR-250N gigabit (1000base-T) router and cat 5e shielded, twisted pair cables. The router supports 1600 Mbit/s according to specifications and, since all the Raspberry Pi computers have 100base-T interfaces, it can in theory service at least 16 of them before any risk of congestion.

4.2.5 Mount and casing



(a) Hardware stack



(b) First assembly of the IIB

Figure 4.6: Shows the hardware stack and the casing.

Both mount and casing are laser cut in 4mm MDF and 3mm acrylic plate. The casing is cut with finger type edge joints (figure 4.6 b) and hot glued on the inside. The front window is attached with clear silicon glue on the inside. This means the front of the casing is free from glue (R-2-2). The back pane is removable and has a hole for cables. The hole is aligned with a door in the cage. The casing is painted black to fit the black inside of the isolator box and to be cleanable (R-2-1).

The hardware is stacked in three layers (figure 4.6 a) with the screen, camera and speaker fixed on the top deck. The camera pi is located below the camera on the middle deck and the output pi on the bottom deck.

4.3 Video

4.3.1 Video conference setup

The first approach was to have one camera next to the screen in a setup similar to video communication on cell phones or on a PC using a webcam. The advantage is that this technique is very simple and cheap to set up. However according to the requirements specification the bird communication might depend on eye-contact (R-4-2), which is impossible in this setup (see figure 4.7).

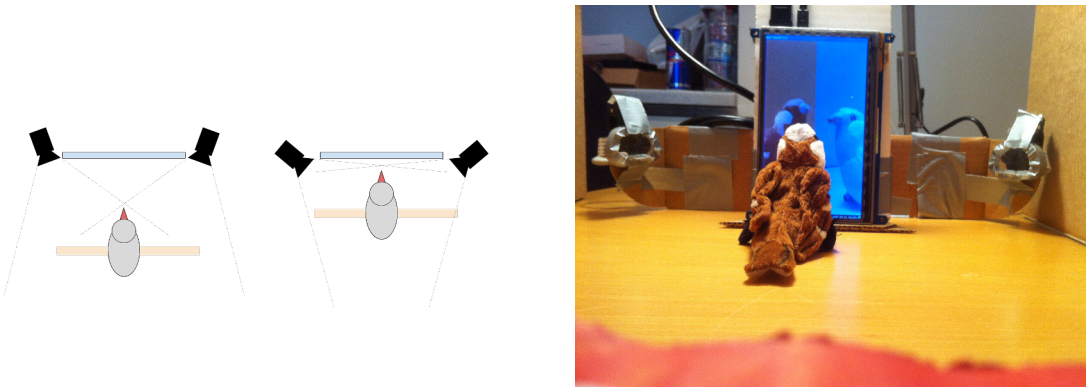
(a) Image with eye-contact (looking into the camera)

(b) Image without eye-contact (looking into the screen)

Figure 4.7: Images of the author (I apologise for the much needed haircut) demonstrating the problem with having a camera next to the screen.

4.3.2 Stereo vision setup

Another approach that was tested was to place the screen between two cameras and use the homography to transform and stitch the images. This technique is well-known in computer vision applications applying stereo cameras. The advantage is that the bird could come very close to the screen and actually touch it, thus opening new possibilities when using a touch screen. The problem is that as the object (the bird) moves closer to the screen, the angle between the cameras must increase in order to avoid occlusion from the screen.



(a) Sketch of the setup seen from above indicating the camera angle issue

(b) The mock-up used to test the setup.

Figure 4.8: The stereo vision setup.

It was concluded from recordings with the mock-up (figure 4.8) and preliminary results from the stitching (figure 4.9) that the distortion was too intense for this to be a viable solution. Additional filtering and interpolation could produce a clearer image, but it would picture a very wide bird with both eyes visible from the front.

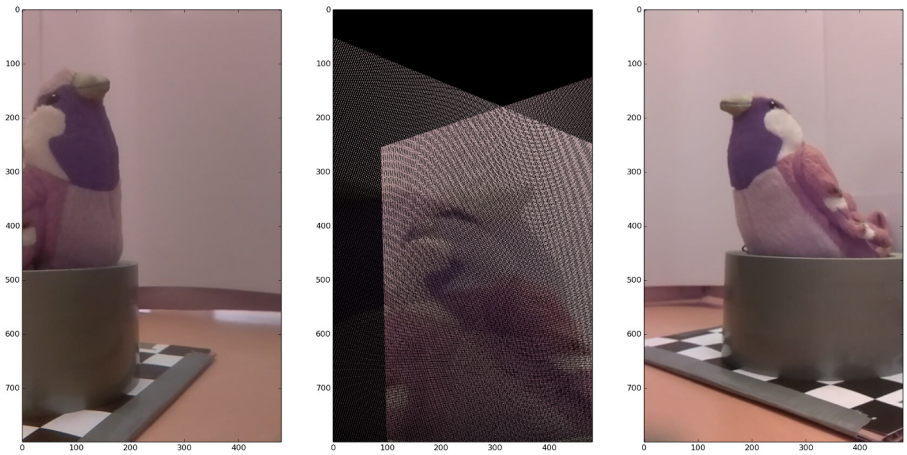


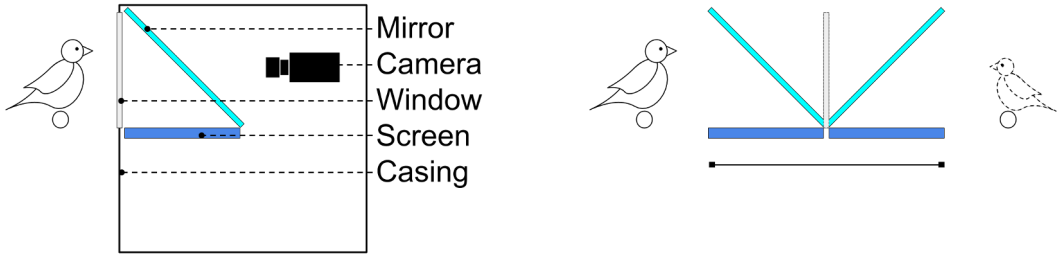
Figure 4.9: The two raw images (left and right) and the stitched image (center).

4.3.3 Teleprompter setup

The chosen approach is a basic teleprompter setup. A teleprompter is a display device that prompts the person in front of the camera with a manuscript. To that person the device looks like a screen, but is really a camera. This way the person appears to be looking into the camera although actually reading from a screen.

The camera is placed behind a slanted one-way mirror and a screen is projecting the image onto the mirror (figure 4.13). The advantage of this approach is that the camera can be placed perfectly in front of the bird as required (R-4-2). The disadvantage is that the setup introduces extra distance between the birds, since the image plane seen by the bird is formed vertically at the end of the mirror. This also introduces a trade-off between the screen size and the perceived distance, since the extra distance is equal to twice the height of the screen (figure 4.13 b). This effect might be countered by some clever zooming, since the birds' depth perception is probably limited as a consequence of monocular vision.

The one-way mirror requires the camera to be in a dark environment and the bird to be in light. This allows the camera to see the bird, while the bird sees the reflection of the screen. The one-way mirror is constructed from a sheet of 3mm transparent acrylic plexiglass coated by 0.02mm silver one-way film with 70% light admittance and 99% reflectance. The mirror is mounted in the casing at a 45° angle right behind the window.



(a) Shows the different parts of the teleprompter setup.

(b) Shows that the perceived distance is the real distance plus double the height of the screen.

Figure 4.10: The teleprompter setup.

4.3.4 Compression

Compression is the process of removing redundant information from data. In still images this is done mainly by encoding areas of similar colour or texture together. In an image with a large portion of blue sky only varying a bit in nuance, it makes little sense to represent each pixel by the combination of four 16-bit colour channels. Instead it would

make sense to name the colour and represent a number of pixels by the named colour plus a small variation. Numerous algorithmic approaches to this exists and each has its own advantages and drawbacks.

In practice video compression is different from compression of still images because most of the redundant information exists between frames. In the present case recording birds most of the image, representing the background, will remain almost the same over consecutive frames. Therefore instead of representing pixels according to named colours, it makes sense to represent changes with respect to a previous frame. Chances are that the changes will be small. This concept of compressing the inter frame temporal redundancy is called inter-frame prediction.

The h.264 (MPEG-4 part 10) compression codec is among the most widely used and is therefore well documented and implemented on all platforms. Apart from that the main reason for choosing h.264 is that it implements the inter-frame prediction feature. The codec designates I-frames (intra-coded) containing all information and P-frames (predictive, inter-coded) encoding the changes with respect to an I-frame. For completeness there is a third frame type called B-frames (bi-directional) but they are irrelevant for live streaming as they aim to reference frames that come later in the stream. It is typical for a codec to implement a range of features for the user to choose from in order to suit many different applications.

A big part of compression algorithms consists of simple computations repeated many times. This is exactly the strength of the Graphics Processing Unit (GPU) and the BCM283x sytem on a chip (SoC) used on Raspberry Pi contains a VideoCore 4 GPU. The GPU supports both encoding and decoding of h.264. The GPU can be accessed through the Open Media Acceleration (omx) framework and is implemented as a pipeline plug-in for gstreamer.

The influence of compressing video shown to birds has not been examined and at a first glance might seem irrelevant. However a technique called chroma sub-sampling takes advantage of the human eye being much more sensitive to differences in luminance than difference in colour. By constructing a colour space consisting of a luminance component and two colour components (YCbCr), the colours can be sub-sampled to a lower resolution than the luminance (figure 4.11) either in the horizontal direction (4:2:0 scheme) or in both directions (4:2:2 scheme). This is almost impossible for the human eye to notice, but could be noticeable to a bird. Since the compression is sufficient even without, chroma sub-sampling is disabled (4:4:4 scheme).

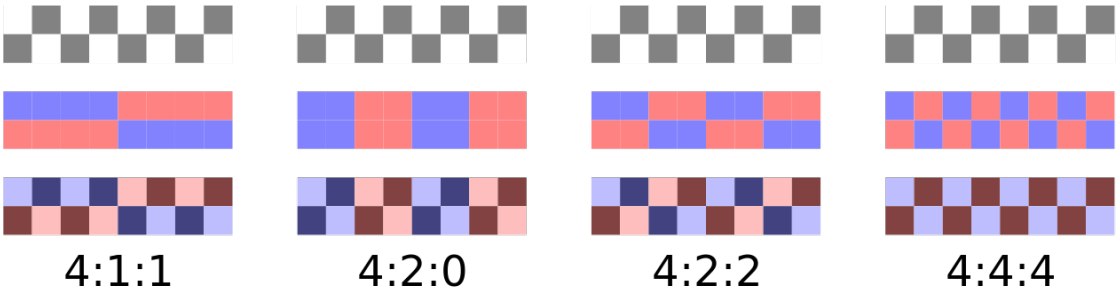


Figure 4.11: Shows the principle of chroma sub-sampling, where luminance and colour are in different resolution (image from wiki-commons).

4.3.5 Streaming

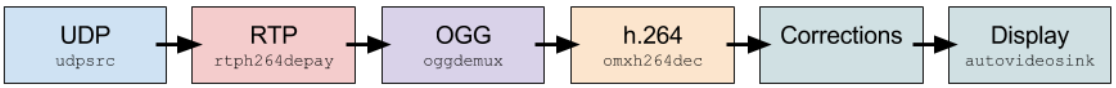
Video streaming can be seen as a pipeline consisting of several steps from the camera driver, over different processing and compression steps to the network driver. This pipeline approach is the basis of the Gstreamer framework used in this project. The idea is that developers write plug-ins for the pipeline. A plug-in is a C-program that takes input from the previous plug-in in the pipe, performs some process and outputs the processed data to the next plug-in. The Gstreamer framework provides data structures for signalling both upstream and downstream between plug-ins, allowing plug-ins to be dynamically reconfigured and to receive information about the stream. It also handles instantiation and tear-down and provides an Application Programming Interface (API). Gstreamer has a large and very active community and a large number of available plug-ins.

The pipeline used to stream data from the cameras consists of several steps (figure 4.12). In the first step (v4l2src) the video4linux driver version 2 (v4l2) is used to capture images from the camera hardware. The v4l2 driver fully supports the Pi camera and thus a large number of settings are available controlling device, frame rate, image size, brightness, sharpness, rotation and h.264 profile.

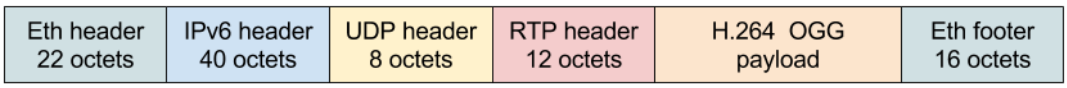
The next step (omxh264enc) encodes the stream in h.264 format. Again this format has a number of possible settings for quantisation of different frame types and interval between Instantaneous Decoding Reset (IDR) frames. Most importantly the default 4:2:2 chroma sub-sampling scheme is disabled. The encoding is executed using the OpenMax library bindings for the embedded Broadcom video4 GPU on the Pi.

The following step (rtph264pay) encodes the stream in real-time protocol (RTP). This is necessary because UDP is used as transport layer protocol and thus implements no sequencing or error detection. The underlying assumption here is that in order to have timely delivery, some packet loss can be tolerated. At thirty frames per second a missing

(a) Combined pipeline at the sender



(b) Combined pipeline at the receiver



(c) Packet format

Figure 4.12: Shows the combined pipeline and packet format for the video streaming.

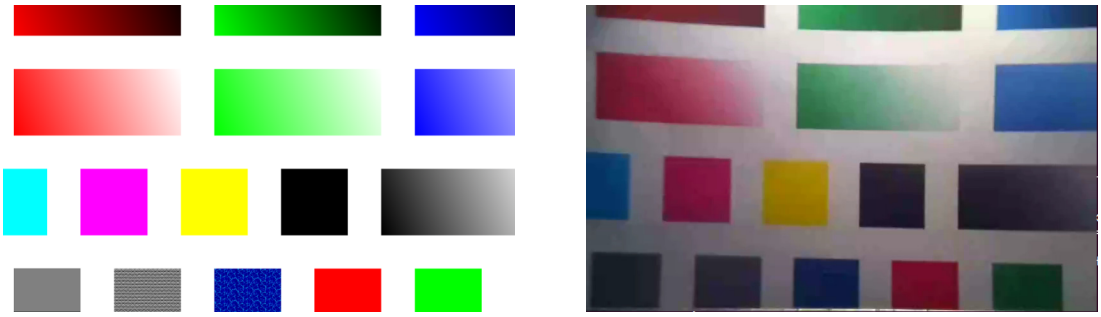
frame is easy to conceal by displaying the previous frame twice. RTP encapsulates the h.264 data and provides timestamps (for synchronisation), sequence numbers (for packet loss detection and reordering) and information about the format of the payload. RTP also provides some control mechanisms communicating off-band at a port number one higher than the RTP connection. The primary function of Real-time Control Protocol (RTCP) is to gather statistics about the quality of the connection. The sender can use these data for adaptive encoding.

The final step (udpsink) is to encapsulate the RTP datagram in a UDP packet and send it to an IPv6 multicast group. At the receiver the corresponding steps are performed in reverse order, first unpacking UDP, then RTP, h.264 decoding and finally a few corrections to the image (flipping, brightness and colour adjustments) before displaying it. The complete pipelines and the format of the packet can be seen in figure 4.12.

4.3.6 Calibration

Due to the reduced light admittance of the one-way mirror, the brightness and colour saturation of the image need to be calibrated. This is done manually with the video balance plug-in when the setup is mounted in the isolator box under the actual lighting

conditions, but a preliminary test was made in advance on the desktop. The calibration is done by placing a colour card in front of the camera in one cage and compare the image projected to the other cage to an identical colour card (figure 4.14). Two different colour cards are used; a real one with samples of different colours and textures and a printed high definition image of zebra finches.



(a) The calibration card

(b) The zebra finch image

Figure 4.13: The original image (left) and the streamed image (right).

Figure 4.14: The video calibration setup with the zebra finch image (printed on both sides).

4.4 Sound

4.4.1 Microphones and speakers

Four microphones are placed in each isolator box and connected to the same 8 channel recorder. The advantage of connecting them to the same recorder is that data from the two setups is synchronised right away. There are two reasons for using multiple microphones. The first reason is so far only theoretical and it is beyond the scope of this report to test it. Assuming that the communicating bird is perched in front of the IIB and that the microphones are positioned at equal distances from that point, averaging the four recorded signals should significantly improve the signal to noise ratio. The explanation is that the signal of interest will reach all four microphones at the same time, whereas noise from reverberations or other birds will arrive at different times. Therefore the signal should be unaffected by the averaging, while the noise should be attenuated. In practice this may require alignment of the signals for example via cross-correlation.

The second reason is that multiple microphones enable sound source localisation based on multi-lateration. The explanation here is that the different arrival times of a signal can be converted to differences in distance. For each microphone pair it is then possible to form a hyperbola and given at least four microphones, the location of the sound source can be estimated. Localising the birds could become an interesting feature at a later stage, although it should be noted that with several birds in the same cage localisation does not

discern between individuals. If the data is to be used in sound source localisation, it would be fairly simple to have all eight microphones in the same box for better precision.

The sound is played back in mono on a 1.5W speaker located inside the IIB in order to protect it (R-2-5). The speaker has a built-in amplifier and is connected to the same computer as the screen. The placement inside the casing will inevitably cause some distortion of the sound. Should further testing address this as a problem, the output (mini-jack) can easily be connected to external speakers.

It is worth noticing that the setup has an inherent risk of feed-back loops, since the sound recorded in one box is played back in the other box, re-recorded there and played back again in the first box. Should this become a problem, there are several ways of dealing with it (microphone and speaker placement, adaptive filtering, frequency shifting etc.). However initial testing showed no problems with feed-back and therefore the problem is not investigated further here.

4.4.2 Sound recording

The sound is recorded using a low cost ultrasonic recording array (URA) developed for research in bio-acoustics (Andreassen et al., 2013). This means that the sound recording is external to the system in compliance with R-1-5 and that the microphones are placed outside the cage so the birds are separated from the microphones (R-2-2). It was considered to include a microphone in the box, but this would require a hole in the box for sound to enter, leading to challenges with dust entering as well as possible pecking in the hole. Some features like sound source localisation might require multiple microphones or a special setup and since most labs already possess expensive sound recording equipment, it was decided to keep it separate.

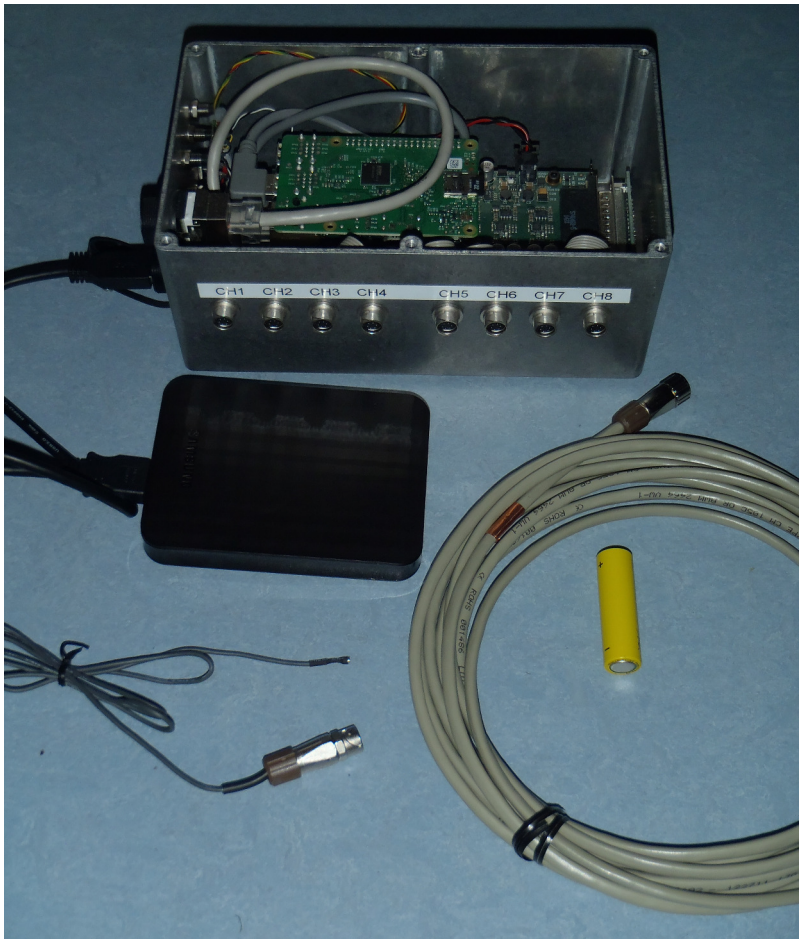


Figure 4.15: The eight channel low cost Ultrasonic Recording Array (URA). The box connected to an external hard-disk (top). Shielded twisted pair wire (right) and the knowles microphone (left).

The URA consists of a closed box with connectors to the eight microphones (figure 4.15). The microphones used in the system are knowles FG23329-PO7 connected via plugs containing a one-wire system with memory for calibration data. Describing this system in detail is beyond the scope of this discussion, but a few important points need to be addressed.

The recording chain consists of a number of analog and digital stages (figure 4.16). The first stage is the microphone and the data sheet states a sensitivity in the relevant range of $-53dB\ re\ 1.0V_{RMS}/0.1Pa$. From the definition of decibel it is possible to find the output per Pa sound pressure:

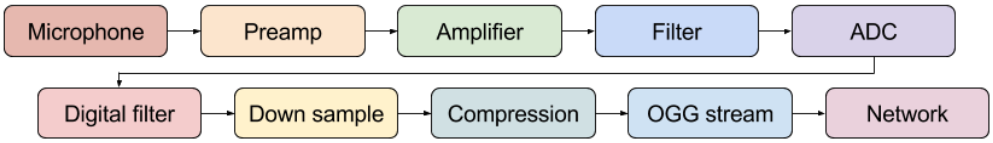


Figure 4.16: Shows the recording chain from the microphone until data is transmitted via network.

$$G = 20log_{10}\left(\frac{V}{V_{ref}}\right) \Rightarrow V = \frac{10^{\left(\frac{G}{20}\right)}}{V_{ref}} \tag{4.3}$$

$$G = \frac{10^{\left(\frac{-53dB}{20}\right)}}{1.0V} = 0.0022\frac{V}{0.1Pa} = 22\frac{mV}{Pa} \tag{4.4}$$

The source level of zebra finch song is approximately $80dB_{RMS}$ re $20\mu Pa$ @ $1m$ (personal communications with Coen Elemans). This corresponds to a sound pressure level at the microphone measured 40cm from the source:

$$80dB_{RMS} + 20log_{10}\left(\frac{1m}{0.4m}\right) = 88dB_{RMS}\ re\ 20\mu Pa\ @\ 1m \tag{4.5}$$

$$10^{\left(\frac{88dB}{20}\right)} \cdot 20\mu Pa_{RMS} = 100,000 \cdot 0.000020Pa_{RMS} = 0.5Pa_{RMS}\ @\ 0.4m \tag{4.6}$$

This means the expected output from the microphone is:

$$0.5Pa_{RMS} \cdot 22\frac{mV}{Pa} = 11mV_{RMS} \tag{4.7}$$

The microphone plug contains a preamp stage of fixed 20gg (26dB) and thus the input to the amplifier stage is:

$$11mV_{RMS} \cdot 20 = 220mV_{RMS} \tag{4.8}$$

The amplifier has a 256 step logarithmically variable gain between 1 and 1000gg (linear in dB from 0dB and 60dB according to data-sheet). The amplifier is followed by a filter stage and finally an analog to digital converter (ADC). The ADC has selectable input range of $\pm 0.5V$ or $\pm 0.75V$, but this corresponds to the maximum peak values so in order to find the correct gain, the RMS value found in (4.8) must be converted to peak values. Assuming pure sine waves this yields:

$$220mV_{RMS} \cdot \sqrt{2} = 311mV_{peak} \tag{4.9}$$

This means the amplifier at input range $\pm 0.5V$ should have a gain of:

$$\frac{0.5V}{0.311V} = 1.6gg(4dB) \tag{4.10}$$

and at input range $\pm 0.75V$:

$$\frac{0.75V}{0.311V} = 2.4gg(7.6dB) \tag{4.11}$$

Since each step of the gain corresponds to roughly $.2dB$ the gain is set to 30 and the input range to $\pm 0.75V$. It is worth noticing that zebra finches are not loud singers and that recording for example canaries would require the URA to implement gain smaller than unity to avoid clipping.

After the amplifier a filter stage can be activated, but is intended for ultrasonic recordings of bats. The filter is band-pass from 9.18kHz to 151kHz, but the 9.18kHz high-pass filter can be disabled, leaving only the 151kHz low-pass filter.

The Nyquist sampling theorem says that the highest frequency component in the signal must be smaller than half the sampling rate. This means the sampling rate must be at least 302kHz and preferably more since the filter is not ideal. The sampling rate is set to 312.500Hz.

The very high sampling rate means a lot of data is produced and thus network congestion could become an issue. The amount of data produced on eight channels is:

$$312.5k\frac{sample}{s} \cdot 16\frac{bit}{sample} \cdot 8channels = 40M\frac{bit}{s} \tag{4.12}$$

The network is ideally 100Mbit/s so it should be possible.

The only currently relevant application that could facilitate such high sample rates is sound source localisation, where the difference in arrival time is used in multi-lateration. If the algorithm for estimating the difference in time of arrival is able to do so within three sample it would mean a precision of:

$$1sample \pm 1sample = \frac{3}{312.5k}s = 9.6\mu s \tag{4.13}$$

Using the speed of sound in air, the time difference can be converted to a difference in distance:

$$9.6\mu s \cdot 340\frac{m}{s} = 3.2mm \tag{4.14}$$

For most practical purposes a sub-centimetre position of the sound source would be unnecessary. In other words there are several reasons to consider reducing the sample rate and a final argument is that most audio formats and applications use standard sampling rates of 44.1k, 48k, 96k or 192k. Re-sampling the signal to 96k would still enable a theoretic precision of approximately one cm for sound source localisation and reduce the network load to 13Mbit/s.

Since the ratio between the two sampling rates is not an integer, asynchronous re-sampling is required to implement this. Asynchronous re-sampling is an advanced topic in signal processing and there are a number of trade-offs in the design of algorithms performing this. Since the re-sampled signal must adhere to a new Nyquist frequency, the signal needs to be low-pass filtered first. Digital filters can be implemented in several ways and are also an advanced topic, but for audio applications, preservation of waveforms is most critical. In filter terms the strategies to obtain this are either linear phase or minimum phase. A filter with linear phase will introduce a delay that is a linear function of the frequency whereas the minimum phase filter is optimised for minimal delay. The sound exchange command-line utility (SOX) implements a linear phase filter in its asynchronous down-sampling functionality and it is used to down-sample each channel to 96k.

If all eight channels were kept together in one stream, the recording system would be easily offloaded, but then the subscriber would need to receive all data although it might need only a portion of it. Another alternative would be to separate it into two topics, each containing the four channels from one isolator box. Finally they could also be separate topics of one channel each. If used for localisation the synchronisation at sample level would be broken by separating the stream in individual channels, but having more than one channel interleaved introduces a need for some sort of key-frame to tell the subscriber where data begins.

For the current application with four microphones in each isolator and not using the data for localisation, it was chosen to publish each channel at a separate topic as it makes most sense that the extra load separating the channels is done by one producer rather than by several consumers.

4.4.3 Compression and streaming

The output from the URA is raw, interleaved, 96k samples/s, 8 channel, signed 16-bit data in big endian. This means the digital filtering and down-sampling is handled by SoX and piped via standard input to a Gstreamer pipeline that handles de-interleaving, compression and network packaging.

Initially the stream is de-interleaved from one eight channel stream to eight one channel streams and then each of the new streams are processed the same way. Each stream is

encoded in vorbis, which is normally a lossy compression format, but in this case it is encoded without compression. The advantage of the vorbis format is that it can encode an arbitrary number of channels in any sampling rate. Most standard formats dictates one or two channels and most have fixed sample rates. The Free Lossless Audio Codec (FLAC) supports 8 channels without loss, but is not supported by the Real Time Protocol.

Next the vorbis stream is embedded in an OGG container, thus adding time stamp and meta-data to the stream (see section 4.1.4 for discussion on the OGG container). Since data is streamed via UDP the real time protocol handles sequencing as described for video in section 4.3.5.

4.4.4 Calibration

It is important to calibrate the equipment in order to know the precision of aquired data and in order to control the stimuli presented to the birds. Each step in the recording chain can be calibrated individually, but a combined calibration gives the best information about the particular setup.

The source level calibration is done in order to know what sound pressure level a number in the output stream corresponds to. The calibration method is to play a one tone test sound and record it with the UAR as well as with a calibrated microphone in the same place. Both IIB, cage and other equipment should be present to get as close to the actual conditions as possible. From the calibrated recording the source level can be found and the output of the UAR can be compared to that. The test could potentially be frequency or position dependent and thus the calibration should be performed at different frequencies and with the sound source in different locations. Alternatively each microphone can be calibrated with a known sound level using a microphone calibrator (figure 4.17). Both methods are used to calibrate the system.

The speakers also need to be calibrated in order to present sound at the same target level as the birds would experience, had they been sitting next to each other. In order to do this the recording chain must first be calibrated. A sound source is placed in one of the boxes in front of the IIB at the location where the bird is expected to be. A 0.1s duration sweep from 500Hz to 8kHz is played back, recorded by the URA, streamed over the network and played back in the other isolator. Comparing the signal recorded in the first box with the signal recorded in the second box should yield similar source levels. Some amount of dampening has to be introduced in order to avoid an infinite feedback loop, but this attenuation is introduced in a controlled fashion and not through poor calibration of the system. It should be noted that the speaker itself has a transfer function that could depend on frequency, but according to its data sheet, the speaker used in the IIB is linear in the 500Hz to 8kHz frequency range.

Figure 4.17: G.R.A.S. sound calibrator.

4.4.5 Alternative equipment

The above discussion was based on the URA system, but one of the requirements for the system was that it should be possible to reuse existing often expensive high quality recording equipment (R-1-5). To exchange the URA with alternative equipment would require it to stream the recorded data to the network. The easiest adaptation would be equipment that is supported by the v4l2 driver or a similar Gstreamer source on linux. Since Gstreamer also supports both windows and Mac OSX, it is likely that a Gstreamer source plug-in already exists. Although adapting the equipment might require a skilled technician, it will in most cases be fairly simple due to the modular structure of Gstreamer.

4.5 User interface

4.5.1 Functionality

The purpose of the user interface is to assist the user in maintaining and controlling the system. The top level use cases include several control procedures, namely starting and stopping the system or parts of it, connecting remotely to a specific computer in the system, monitoring the current video feed and accessing the system log. To make this more tangible example commands would be:

```
ats all start
```

```
ats box1cam poweroff
ats box2output ssh
```

The command interface follows the pattern:

```
ats [OPTIONS] <unit name>|all [COMMAND]

OPTIONS
-d       Dry run. Echoes commands instead of running them.
-v       Verbose. Outputs all information.

COMMANDS
ssh      Connect via ssh
net      Setup internet gateway
start    Start unit
stop     Stop unit
nop      Do nothing
```

The control computer (CC) is typically a laptop temporarily connected to the ATS network. Once the system is running, the CC can be removed, since it is only needed when the system is reconfigured or for monitoring. The service modules in the system on the other hand are expected to be present at all times as other modules might depend on data from them.

The CC contains all scripts and configuration files for the entire system. This centralises the maintenance of the system and thus accommodates the disadvantage of distributed systems that maintaining many small machines is more cumbersome than maintaining a big one. The scripts are executed in a shell environment and therefore exerts no demands on programming language, available libraries or access to hardware. This complies with requirements R-1-1 and R-1-2.

A control procedure is executed on the control computer and comprises one or more control tasks executed on the service module. Any procedure begins with setting up the shell environment to correspond to the current service module. The files relevant to the procedure are then transferred to the service module and finally the commands are executed. This process is repeated for each of the service modules involved in the procedure (figure 4.18).

The output from the terminal in which the script is run is multiplexed to a local log file and sent to a common syslog server topic (a designated IPv6 address). If a syslog server has subscribed to the topic, it will receive log updates from all computers on the network as well as from the router. If no server is present, the packets are dropped and only the

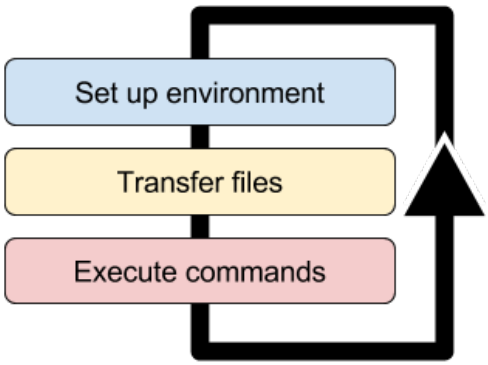


Figure 4.18: Shows the tasks that are executed on the control computer. The tasks are repeated for each of the service nodes involved in the procedure.

local copy of the log file holds the entry.

4.5.2 Monitoring tools

The monitoring tools consist of two parts. The administration interface is a standard Unix command-line user interface (CLUI) and consists of a back-end based on scripts that are executed on the machines in the system and a graphical user interface (GUI) built on top of that.

The administration interface is not based on use cases, since this interface is for the expert user who needs full accessibility for developing and extending the system. The CLUI contains a set of tools to make things easier for the expert user and is based on the use cases in table 4.1

The GUI is developed primarily for the end user and is based on the use-cases for the CLUI as well as general monitoring possibilities for example, live images and sound from the cages or previously recorded data. It also provides the user with an interface to the features extracted from the data. A proof of concept was developed in Qt from which it is possible to run scripts (with arguments) and receive standard output from them. It can also show video streams, but it should be noted that in practice Qt sends the coordinates of the window to Gstreamer that will then overwrite that part of the frame buffer directly. This is of course the fastest possible solution, but it also means that the image is never passed to Qt so any manipulation must be done within the Gstreamer pipeline.

Use case	**Scope**	**Command**
Start unit	All units	`start`
Stop unit	All units	`stop`
Connect to unit	All units	`ssh`
Setup interenet on unit	All units	`net`
Shutdown unit	All units	`poweroff`
Turn volume up/down	Output units	`volume [<value >|up|down|mute|unmute]`
Test speaker	Output units	`sptest`
Set screen balance	Output inits	`screen [hue <value>|sat <value>|br <value>]`
Test screen	Output units	`sctest`
Show video stream	Camera units	`show`

Table 4.1: Shows the use cases of the system.

Chapter 5

Validation

Not all tests have been performed at the deadline of this thesis. The reason is that the the Department of Biologi is currently moving and the isolator boxes have been used in another experiment up until that, so they have not been available.

5.1 Design inspection

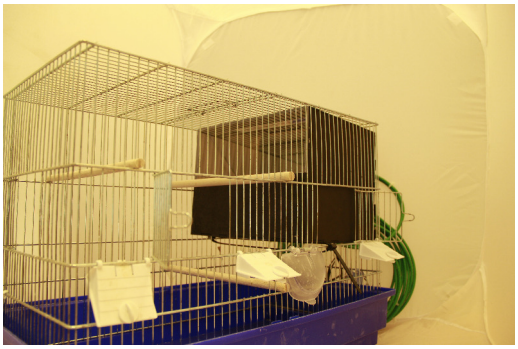
Several of the requirements concern the design choices made during development of the system and most have been addressed in the implementation chapter.

Regarding the general design requirements the modules in the system are only dependent on data (R-1-1) and clearly follow the publish-subscribe pattern (R-1-2). Furthermore the system is distributed and all communication between the modules is via network (R-1-6). The software developed in this project is open sourced under the GNU General Public License (GPL) and Gstreamer is licenced under the GNU Lesser General Public License (LGPL). This means all software in the project is open source (R-1-3). Although this work is based on the URA for sound recording, there are no hardware dependencies on it and thus other equipment could be adapted (R-1-5). All parts of the system are calibrated in order to provide the accuracy of measurements (R-1-7).

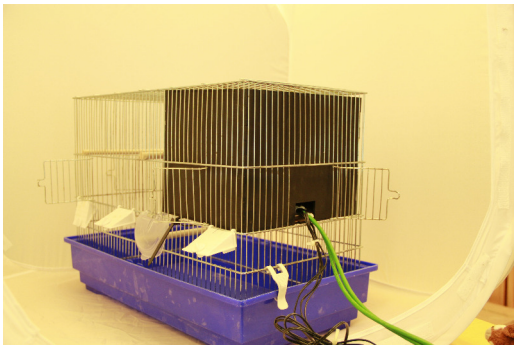
Regarding the birds, all surfaces presented to them are painted (R-2-1) and have no peckable parts (R-2-2). The cages have 40cm x 60cm floor equal to $2.400cm^2$, enough for 4 birds (R-2-3). Since the IIB is fixed to the top part of the cage, the bottom part stays removable for practical purposes (R-2-4). All equipment except wires is concealed in the IIB and is thus protected (R-2-5).

The teleprompter design ensures mediation of eye contact (R-4-2) and the system has

been mounted in the cages (figure 5.1) thus meeting requirement R-5-2. The IIB cables fit through the hole in the isolator box (R-5-1), but with eight microphones the URA does not. Instead of widening the hole, those cables have been wired through the air intake. Finally all data is treated as streams, thus meeting requirement R-6-3.



(a) From the front



(b) From the rear

Figure 5.1: The IIB mounted in the cage.

5.2 Cost analysis

Since the recording equipment is considered external, the only cost of the system is the IIB and some network equipment. The parts used in the IIB are summarised in table 5.1 and total roughly DKK 1.600. The casing is not included since the materials are very low cost, but the equipment necessary to produce it is quite costly and trying to guess a fair price for the casing is therefore difficult. Add to that the D-link DSR-250N costing another DKK 1.600 and the total reaches DKK 4.800 which is barely below the required maximum price of DKK 5.000 (R-1-4). It is important to note that this cost does not include salary for a technician to assemble the parts.

#	**Item**	**Supplier**	**Price**	**Price in DKK**
1	Raspberry Pi	RS		188,90
1	Pi Camera v2	RS		188,44
3	Glass	Tiger		75,00
1	Raspberry Pi 2	RS		251,89
1	Speaker	Fona		99,95
1	Screen	Ebay	£ 39.99	384,60
1	HDMI cable	RS		60,27
3	5V Power supply	RS		150,00
2	SD cards 8G	RS		265,00
2	Cat 5e cable	RS		150,00
Total				1.625,15

Table 5.1: Cost of the IIB.

5.3 Robustness

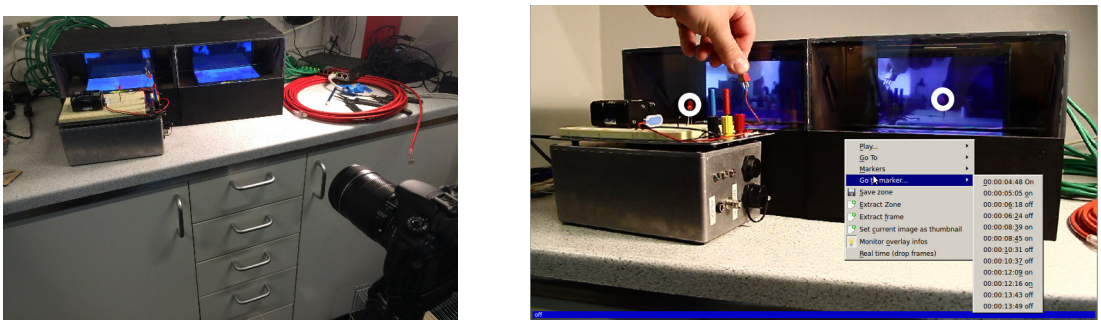
Robustness against failing power supply is tested simply by shutting off power to the system and checking that everything comes back on-line as expected when power is restored. This test should be performed once the complete system is installed. At the moment each component in the system has been tested and all pass the test (R-6-5).

The other part of the robustness test was to keep the system running for a week, which it also passed (R-6-4)

5.4 Video tests

5.4.1 Video delay

In order to estimate the delay in the visual system, a simple setup with an LED connected to a switch is used (figure 5.2). The LED is placed in front of one of the IIBs and an external camera is used to record a video showing both the LED and the screen of the other IIB. When turning the LED on, the number of frames until it turns on in the IIB can be counted.



(a) The test setup

(b) Results from the annotation

Figure 5.2: The video delay test.

One video was recorded, in which the LED is turned on and off three times. The video was recorded with a Canon EOS 60D camera at 50 frames per second. The video was manually annotated in kdenlive. The criterion used was to count from, but not including, the first frame where the direct view of the LED changes until and including the first frame where the indirect view of the LED changes. The resulting annotations from kdenlive are shown in figure 5.2. All six events had a delay of six frames except one that was seven frames. At 50 fps this means the maximum measured delay was:

$$dt = \frac{7}{50}s \pm \frac{1}{50}s = 140ms \pm 20ms \tag{5.1}$$

The result is close to the limit for low latency of 150ms set in R-6-1. In fact the resolution of the measurement is too low to tell if the requirement holds. If a better estimate is desired the test could be improved by generating a delta autocorrelation random sequence and cross-correlating the two video streams for sub-sample accuracy. The conclusion for this test is that the system passes, although barely.

5.4.2 Flicker

It is paramount that the screens do not flicker and to test this, a slow motion recording with an iPhone 6 at 120 fps was made. At this rate it shows no flicker (R-4-1)

5.5 Sound tests

5.5.1 Sound delay

When both systems are mounted in the isolators, a sound is played in one of the speakers. The sound is recorded both in the box where the sound is played and in the other box where it is transmitted through the system and played back in the speaker. Since both signals are recorded by the same URA, the delay can be estimated from comparing the recorded signals. This test has not yet been carried out.

5.5.2 Homogeneous sound field

In order to test the physical properties of the isolator boxes, two tests must be made. First the empty box is tested using calibrated high grade equipment and next the box was measured while containing all the equipment expected to be there in the real trials. The basic idea is to measure the transfer function of the speaker and microphone as a black box. This is done in an anechoic room at 1m distance (figure 5.3). Next the same measurement is repeated inside the box and any difference in transfer function must be due to the box.

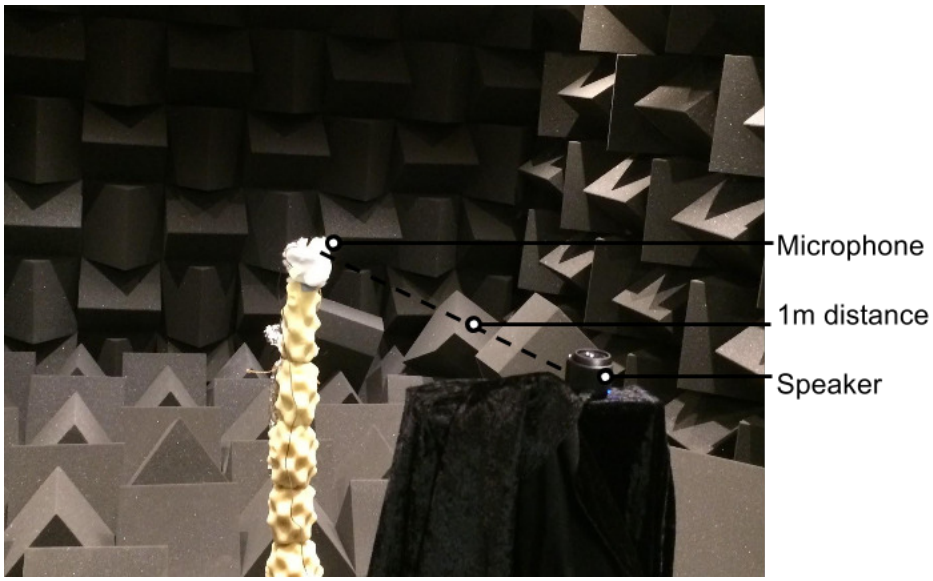


Figure 5.3: The test setup in the anechoic room.

The purpose of the first test is to test the physical properties of the isolator itself. The

procedure consists of a series of measurements with a calibrated microphone at a fixed location and a calibrated sound source at discretely varied locations.

The purpose of the second test is to reproduce the first test, but in a setup as close to the actual recording conditions as possible. In this test the isolator contains the cage and IIB, the recordings are made with the actual recording equipment and the location of the calibrated sound source is only varied within the cage, focusing mostly on the perch in front of the camera.

In both tests a series of twenty sweeps are played back and since the characteristics of the sound source and the microphone are known, the transfer function of the box can be deduced. Since the transfer function is a function of the sound source and microphone locations, the measurement is repeated for each location.

This test has been prepared, but has not yet been carried out (R-3-2). Another test is to analyse the data for reverberations and this has also yet to be carried out (R-3-3).

5.5.3 Isolation

In order to quantify the sound isolation between the isolator boxes, a measurement of the crosstalk was made. To get a realistic although less general measurement, it was made with live zebra finches in one box while recording from that and the empty box next to it.

The recordings were made simultanously in both boxes, using Dayton Audio Electret Measurement Microphones (EMM-6) connected to a Roland Octa-capture Hi-Speed USB Audio Interface. Capturing was done with Sound Analysis Pro software (Tchernichovski et al., 2000) and subsequent data analysis in MATLAB.

The requirement was set to at least a 40dB attenuation in the range 500Hz-8kHz. The plot in figure 5.4 shows the power spectral density estimate of a one minute recording. On the plot a black dashed line box marks on the frequency axis the range 500Hz-8kHz and on the power/frequency axis the range from the peak of the red graph and 40dB below the peak. This means the sound has been sufficiently attenuated by the isolator box and thus requirement R-3-1 is met.

5.5.4 Fan noise

The self noise of the system is recorded when everything is mounted in the isolators. This test has not yet been carried out.

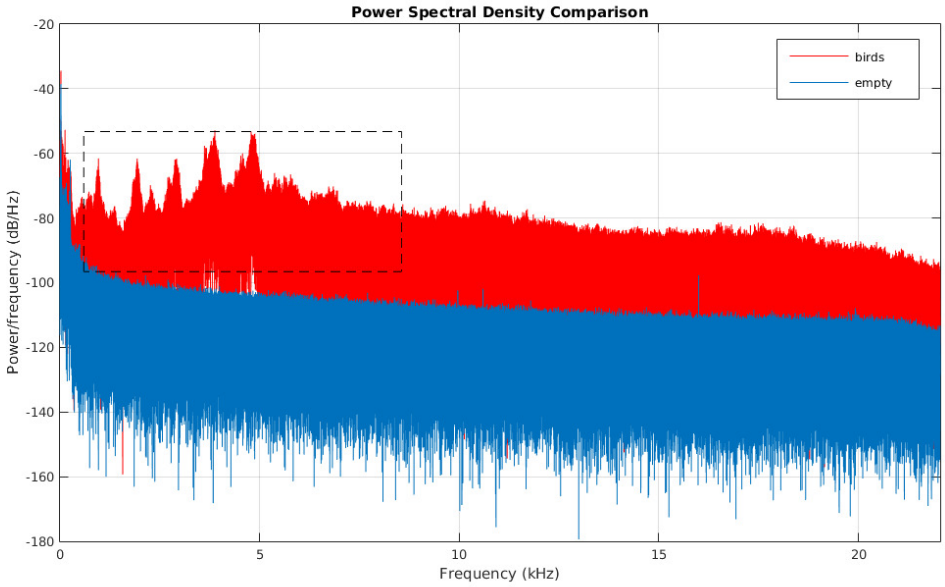


Figure 5.4: Comparison between the power spectral density of the two recordings. The box shows 40dB attenuation in the audible range of the zebra finch (500Hz-8kHz).

Chapter 6

Conclusion

A system for artificial tutoring of songbirds was developed based on thorough analysis of both biological and technical issues. The real test will be if the birds actually believe it and if they are able to interact via the system, but that test is beyond the scope of this thesis. Instead the requirements for the system were tested and the implementation passed or nearly passed all of them. A few tests have not yet been performed, but in general the technical elements of the setup works.

What is much more interesting are the flaws in the system and how to improve on them. A potentially big problem that many papers mention is reflections. When experiments are run where the birds communicate through a glass window, reflections seem to cause at best a change in behaviour and even worse they may cease to communicate. If it is the case, much more expensive anti-reflection treated glass must be used for the windows in the IIBs.

Another possible issue is the lighting in the isolator box. Due to the reduced light intensity caused by the mirror, the setup performs poorly in low light conditions. When the lighting is insufficient it causes the camera sensor to integrate more and the image becomes very grainy. If this turns out to be a problem, additional light might have to be mounted either in the IIB, illuminating the bird through the window or from the outside of the cage.

The synchronisation of auditory and visual stimuli could also become a problem. It was indirectly assumed in the design that birds are inferior to humans in detecting such time differences, but that might not be true at all. It has been shown that humans are able to counter the differences: for example, when listening to people on a stage far away, we compensate for the difference between the travelling times of light and sound respectively (Murray and Wallace, 2011). In fact, when taking into consideration the different propagation times in the central nervous system, we only actually experience synchrony when talking to someone 15m away, the so called 'horizon of simultaneity' (Pöppel and Artin,

1988). There are at least four theories about how this works in humans, but the birds might not do it at all and thus they could be much more sensitive to asynchrony.

At a more technical level there is a difference between how the developers at D-link interprets IGMPv3 snooping functionality and how I interpret it. The RFC3376 clearly describes that multicast traffic should only be transmitted on interfaces that explicitly joined the group. The D-link DSR-250N per default broadcasts packets to all interfaces that have not specifically left the group, which inevitably leads to network congestion. If they are not willing to update the implementation, the firmware could be upgraded to an open source version, but at the current network size broadcasting is not a problem.

A final concluding remark is that the system is still a work in progress and many more challenges, flaws, imperfections and surprises await. Several investigations will be made possible once the birds are actually in the cage like clever zooming and cropping of the image, testing the possibilities of noise reduction in the setup with four microphones and looking into sound source localisation.

In the next phase of the project, the system will be used for trials with zebra finches. Since biologists will have to use the system, the GUI needs to be further developed in collaboration with the users. In the early trials a module for data collection needs to be developed implementing methods for recording, playback and seeking.

Code for this project is open source and available from: `https://github.com/LeonBondeLarsen/ATS`

Bibliography

Patrice Adret. Discrimination of video images by zebra finches (Taeniopygia guttata): Direct evidence from song performance. *Journal of Comparative Psychology*, 111(2): 115–125, 1997. ISSN 0735-7036. doi: 10.1037/0735-7036.111.2.115.

Zachary Aidala, Leon Huynen, Patricia L R Brennan, Jacob Musser, Andrew Fidler, Nicola Chong, Gabriel E Machovsky Capuska, Michael G. Anderson, Amanda Talaba, David Lambert, and Mark E. Hauber. Ultraviolet visual sensitivity in three avian lineages: Paleognaths, parrots, and passerines. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 198(7):495–510, 2012. ISSN 03407594. doi: 10.1007/s00359-012-0724-3.

N. Amin, A. Doupe, and F. E. Theunissen. Development of selectivity for natural sounds in the songbird auditory forebrain. *J Neurophysiol*, 97(5):3517–31, 2007. ISSN 0022-3077 (PRINT). doi: 10.1152/jn.01066.2006.

Aaron S Andalman and Michale S Fee. A basal ganglia-forebrain circuit in the songbird biases motor output to avoid vocal errors. *Proceedings of the National Academy of Sciences of the United States of America*, 106(30):12518–23, 2009. ISSN 1091-6490. doi: 10.1073/pnas.0903214106.

Tórur Andreassen, Annemarie Surlykke, John Hallam, and David Brandt. Ultrasonic recording system without intrinsic limits. *The Journal of the Acoustical Society of America*, 133(6):4008–18, 2013. ISSN 1520-8524. doi: 10.1121/1.4802891.

Luis F. Baptista and Lewis Petrinovich. Song development in the white-crowned sparrow: social factors and sex differences, 1986. ISSN 00033472.

Andrew T. D. Bennett, Innes C. Cuthill, Julian C. Partridge, and Erhard J. Maier. Ultraviolet vision and mate choice in zebra finches. *Nature*, 380(6573):433–435, 1996a. ISSN 0028-0836. doi: 10.1038/380433a0.

Andrew T D Bennett, Innes C Cuthill, Julian C Partridge, and Erhard J Maier. Ultraviolet vision and mate choice in zebra finches. *Nature*, 380(6573):433–435, 1996b.

K. Birman and T. Joseph. Exploiting virtual synchrony in distributed systems. *ACM SIGOPS Operating Systems Review*, 21(5):123–138, 1987. ISSN 01635980. doi: 10.1145/37499.37515.

Barbara A. Caspers, Joseph I. Hoffman, Philip Kohlmeier, Oliver Krüger, and E. Tobias Krause. Olfactory imprinting as a mechanism for nest odour recognition in zebra finches. *Animal Behaviour*, 86(1):85–90, 2013. ISSN 00033472. doi: 10.1016/j.anbehav.2013.04.015.

Dennis Eckmeier, Bart R H Geurten, Daniel Kress, Marcel Mertes, Roland Kern, Martin Egelhaaf, and Hans Joachim Bischof. Gaze strategy in the free flying zebra finch (Taeniopygia guttata). *PLoS ONE*, 3(12), 2008. ISSN 19326203. doi: 10.1371/journal.pone.0003956.

Herbert Gross. *Handbook of opticl systems*. 2005. ISBN 978-3-527-40377-6.

N S Hart, J C Partridge, I C Cuthill, and a T Bennett. Visual pigments, oil droplets, ocular media and cone photoreceptor distribution in two species of passerine bird: the blue tit (Parus caeruleus L.) and the blackbird (Turdus merula L.). *Journal of comparative physiology. A, Sensory, neural, and behavioral physiology*, 186(4):375–387, 2000. ISSN 0340-7594. doi: 10.1007/s003590050437.

Souleiman Hasan, Sean O'Riain, and Edward Curry. Approximate Semantic Matching of Heterogeneous Events. *6th ACM International Conference on Distributed Event-Based Systems (DEBS 2012)*, pages 252–263, 2012. ISSN 1450313159. doi: http://dx.doi.org/10.1145/2335484.2335512.

S Hunt, Ic Cuthill, Jp Swaddle, and Atd Bennett. Ultraviolet vision and band-colour preferences in female zebra finches, Taeniopygia guttata. *Animal behaviour*, 54:1383–92, 1997. ISSN 0003-3472. doi: 10.1006/anbe.1997.0540.

Maki Ikebuchi and Kazuo Okanoya. Male Zebra Finches and Bengalese Finches Emit Directed Songs to the Video Images of Conspecific Females Projected onto a TFT Display. *Zoological Science*, 16(1):63–70, 1999. ISSN 0289-0003. doi: 10.2108/zsj.16.63.

ITU. Recommendation {ITU-R BT.1359-1}: Relative timing of sound and vision for broadcasting. pages 1–5, 1998.

Anthony Leonardo and Masakazu Konishi. Decrystallization of adult birdsong by perturbation of auditory feedback. *Nature*, 399(6735):466–470, 1999. ISSN 0028-0836. doi: 10.1038/20933.

Dina Lipkind, Gary F Marcus, Douglas K Bemis, Kazutoshi Sasahara, Nori Jacoby, Miki Takahasi, Kenta Suzuki, Olga Feher, Primoz Ravbar, Kazuo Okanoya, and Others.

Stepwise acquisition of vocal combinatorial capacity in songbirds and human infants. *Nature*, 498(7452):104–108, 2013.

Micah M Murray and Mark T Wallace. *The neural bases of multisensory processes*. CRC Press, 2011.

OECD. Test No. 223: Avian Acute Oral Toxicity Test. *OECD Guidelines for the Testing of Chemicals*, 1(July):25, 2010. doi: 10.1787/9789264090897-en.

Ernst Pöppel and Tom Trans Artin. *Mindworks: Time and conscious experience.* Harcourt Brace Jovanovich, 1988.

Morgan Quigley, Ken Conley, Brian Gerkey, Josh FAust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Mg. ROS: an open-source Robot Operating System. *ICRA*, 3(Figure 1):5, 2009. doi: http://www.willowgarage.com/papers/ros-open-source-robot-operating-system.

Christian Schloegl, Kurt Kotrschal, and Thomas Bugnyar. Gaze following in common ravens, Corvus corax: ontogeny and habituation. *Animal Behaviour*, 74(4):769–778, 2007. ISSN 00033472. doi: 10.1016/j.anbehav.2006.08.017.

Judith Schmidt, Christelle Scheid, Kurt Kotrschal, Thomas Bugnyar, and Christian Schloegl. Gaze direction - A cue for hidden food in rooks (Corvus frugilegus)? *Behavioural Processes*, 88(2):88–93, 2011. ISSN 03766357. doi: 10.1016/j.beproc.2011.08.002.

Peter J B Slater, L A Eales, and N S Clayton. Song learning in zebra finches (Taeniopygia guttata): Progress and prospects. *Advances in the Study of Behavior*, 18:1–34, 1988.

O Tchernichovski, F Nottebohm, Ce Ho, B Pesaran, and Pp Mitra. A procedure for an automated measurement of song similarity. *Animal behaviour*, 59:1167–1176, 2000. ISSN 0003-3472. doi: 10.1006/anbe.1999.1416.

Advanced Television and Systems Committee. ATSC Implementation Subcommittee Finding : Relative Timing of Sound and Vision for Broadcast. (June), 2003.

Dana Troxel. Understanding Acoustic feedback suppressors. *Rane Corporation*, 2005.

Evren C Tumer and Michael S Brainard. Performance variability enables adaptive plasticity of 'crystallized' adult birdsong. *Nature*, 450(7173):1240–4, 2007. ISSN 1476-4687. doi: 10.1038/nature06390.

Audrey C. Younkin and Philip J. Corriveau. Determining the amount of audio-video synchronization errors perceptible to the average end-user. *IEEE Transactions on Broadcasting*, 54(3):623–627, 2008. ISSN 00189316. doi: 10.1109/TBC.2008.2002102.

Richard A Zann and Michael Bamford. *The Zebra Finch - A Synthesis of Field and Laboratory Studies*. Oxford University Press, Oxford, 1996. ISBN 0198540795.