

BODY SYNTHESIZER

AALBORG UNIVERSITY ESBJERG
SPRING 2003

AUTOMATIC

Abstract

In this report we will describe how we created the “Body Synthezeiser”.

The “Body Synthezeiser” is a system that enables the user to generate various sounds by the mean of body movements.

Main tool in this project is the open source program EyesWeb. By creating and combining patches we made it possible to convert different data types into midi messages.

Keywords: Body movements, synthezeiser, EyesWeb, Midi, Theme room

1	INTRODUCTION	3
1.1	THE INITIAL IDEA	3
1.2	DEFINITIONS	3
1.2.1	THE ROOM	4
1.2.2	TOOLS	5
1.2.3	PROBLEMS	6
1.3	STAGES IN THE DEVELOPMENT OF THE PROTOTYPES	6
1.4	GOALS & EXPECTATIONS	7
1.5	DELIMITATIONS	7
2	THEORETICAL BACKGROUND	8
2.1	STATE OF ART	8
2.2	THE EYESWEB PROJECT	8
2.2.1	SOUNDBEAM™	9
2.2.2	BIGEYE	9
2.2.3	PALINDROME	10
2.2.4	SIMILARITIES / DIFFERENCES	11
2.3	DEFINITION OF A DIGITAL IMAGE	11
2.3.1	GREYSCALE IMAGES	11
2.4	COLOUR SPACES	12
2.4.1	THE RGB COLOUR SPACE	12
2.5	THRESHOLD	12
2.6	DIGITAL IMAGE PROCESSING FILTERS	12
2.6.1	CONVOLUTION	13
2.6.2	LINEAR FILTER	13
2.6.3	NON-LINEAR FILTER	14
2.7	MORPHOLOGICAL OPERATIONS	15
2.7.1	DILATE & CLOSE OPERATIONS	16
2.8	LOGICAL OPERATIONS	17
2.8.1	MONADIC OPERATION: NOT	17
2.8.2	DYADIC OPERATION: AND	18
3	EXPERIMENTAL SETUP	19
3.1	PROTOTYPE 1: MIDI TEST	19
3.1.1	OBJECTIVE	20
3.1.2	DESCRIPTION	21
3.1.3	PROBLEMS	21
3.1.4	SOLUTIONS	21
3.2	PROTOTYPE 2: WAVE TEST	21
3.2.1	OBJECTIVE	23
3.2.2	DESCRIPTION	23
3.2.3	PROBLEM	23
3.2.4	SOLUTION	23

3.3	PROTOTYPE 3: TONE TEST	23
3.3.1	OBJECTIVE	24
3.3.2	DESCRIPTION	24
3.3.3	PROBLEM	25
3.3.4	SOLUTION	25
3.4	PROTOTYPE 4: RED HAND	25
3.4.1	OBJECTIVE	26
3.4.2	DESCRIPTION	27
3.4.3	PROBLEM	27
3.4.4	SOLUTION	27
3.5	PROTOTYPE 5: COMPLETE RED HAND	28
3.5.1	OBJECTIVE	30
3.5.2	DESCRIPTION	31
3.5.3	PROBLEM	32
3.5.4	SOLUTION	32
3.6	PROTOTYPE 6: CHOICE PROJECT	32
3.6.1	OBJECTIVE	33
3.6.2	DESCRIPTION:	33
3.6.3	PROBLEMS	35
3.7	PATCH CONCLUSION	35
4	<u>THE EXPERIMENTS</u>	<u>36</u>
4.1	THE SETUP OF THE ROOM	36
4.1.1	DESCRIPTION	36
4.2	PRELIMINARY VIDEO FOOTAGE	36
4.3	REAL TIME TEST	37
5	<u>SUMMARY AND DISCUSSIONS</u>	<u>38</u>
5.1	PROBLEMS DURING THE EXPERIMENTS	38
5.1.1	PRELIMINARY VIDEO RECORDINGS	38
5.1.2	REAL-TIME EXPERIMENTS	38
5.2	DID WE ACHIEVE OUR GOALS	39
5.2.1	TRANSFORMATION OF MOVEMENT INTO SOUND	39
5.2.2	THE THEME ROOM	39
5.2.3	INTERACTION	39
5.3	REAL-TIME BODY SYNTHESIZER VS. RECORDED FOOTAGE	39
5.4	VIDEO CAMERA CONTROL VS. ULTRA-SOUND (SOUNDBEAM™)	40
5.5	VISIONS WITH THE FURTHER DEVELOPMENT	40
5.6	VISIONS WITH THE USAGE OF THE END-PRODUCT	40
6	<u>CONCLUSION</u>	<u>41</u>
7	<u>LITERATURE/RESOURCES</u>	<u>41</u>

1 Introduction

This chapter will introduce the reader to the ideas behind the project Body Synthesizer, also called Body Synth.

We will uncover the definition of the Body Synth: the room, the tools we used and the problems we encountered during the project.

We will categorize the various stages in the development of the prototypes and the goals of the project will also be stated in this introduction.

In the process of realising this project we found it necessary to adjust our criteria of success and write a section describing the delimitation of the project, this and the expectations of the project will be described at the end of this chapter.

1.1 The initial idea

When a person enters the room he or she instantly activates a sound source¹. This happens continuously by changing the soundscape with a variety of movements. This is an intriguing thought and a cornerstone of the project.

With inspiration found in SOUNDBEAM™ and other projects, we will with the use of a digital video camera, experiment and find different ways of transforming body movements into the creation and manipulation of sound, hence body synthesizer².

For the purpose of isolating the different body movements, i.e. movements by hands and feet, the subject will be wearing different kind of coloured clothing.

1.2 Definitions

In this section we will briefly describe the room, the tools and the problems we expect to encounter.

¹ See short description in the next section or in details in chapter 4

² See chapter 2

1.2.1 The room

We want to create a room where participants will touch and be touched by sound.

We see the room as a virtual room, a place where people can explore and play the room within the room: A room you hear rather than see; a room you change by the mere fact that you are a part of it. We will categorize the room in 3 different levels:

- 1) In this level we will work with a video frontal view of a subject and features provided by the open source program EyesWeb³. We will isolate the data created by body movements of the subject. In a further development of this level we want to use these isolated data to trigger wave samples, generate sinusoidal tones or generate midi sounds. It is essential for the success of the project that the subject has a certain control of the perceived sound.
- 2) This version is a further development of level one. At this stage the room has a sound that will be generated as soon as a subject enters the room. The sound will change according to where the subject stands and what part of the body the subject moves. It is intended to create the illusion that you can manipulate the sound by moving the mass of air in the room. E.g. velocity or volume can be linked to the vertical movement of a hand; the closer the hand gets the ceiling the louder the sounds gets. Midi generated sound in conjunction with wave samples will be used to generate a more complex sound experience.
- 3) At this level the subject will be able to create sounds within a theme room. We want to give the user an experience of being in a particular place. The place could be a big city, the ocean and so forth. The soundscape will be created with theme related samples being triggered and manipulated in

³ See section 1.2.2

various ways in various parts of the Room. The difference in this room, compared to the previous versions, is that you will be on an exploratory sound journey that will trigger your imagination rather than playing the room as a virtual instrument.

Physically the room is thought of as a square where the walls and the floor are covered with black cloth. In the room there will be placed a computer connected to a digital video camera (see figure 1.1). As mentioned in the following section we might have to create a set up with more computers in order to make the soundscape work in real time without any noticeable delay.

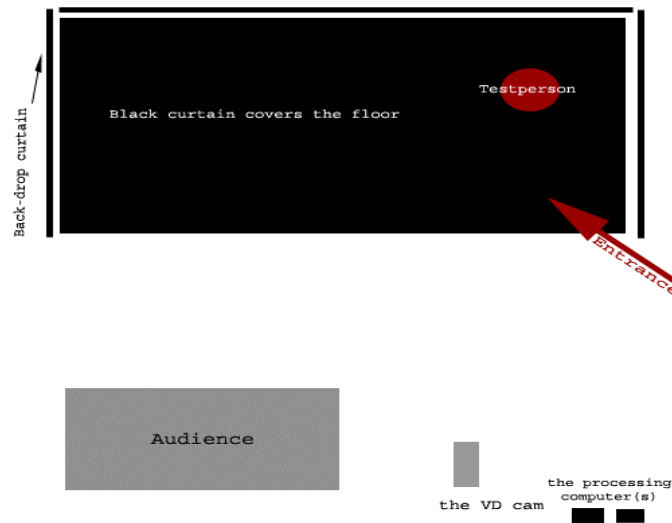


Figure 1.1: The room at the pre-experiment stage

1.2.2 Tools

We use EyesWeb to extract data of movements from a person in the room and process the data into sounds. The main source of the program was made in Italy at DIST – University of Genoa – Laboratorio di Informatica Musicale – Infomus⁴. The program has the ability to extract (almost) any kind of data from an input, visual as well as audio.

⁴ <http://www.infomus.dist.unige.it>

EyesWeb can be integrated with other modules and sensor systems. It supports communication with MIDI, fast serial port, and a proprietary socket software library.

In our experimental work we will also use Premier 6.5 for editing video footage.

1.2.3 Problems

In this project we face two potential problems.

First, we have limited knowledge about the EyesWeb software we will use. To solve this problem we will use the EyesWeb newsgroups and consult people who have experience with the software.

Second, we most likely will encounter problems with large amount of data processing and limited computer resources. The solutions to this problem will be covered in the third chapter: *The preceding experimental setup*.

1.3 Stages in the development of the prototypes

Our first look at EyesWeb will be to explore what exist in the field of tracking body movements.

From this starting point, we will develop our preliminary prototype patch, which will track the red hand of a person in movement. This video footage is known as “micro-dance.avi”. After succeeding in extracting the red hand of the dancing person, we will connect the data from the movement of the hand to a simple tone generator. Following this it should not be difficult to connect a midi output to the patch. In between these prototypes we will experiment with some sound patches.

After experimenting with the “micro-dance.avi”, our next step is to make our own footage of a person wearing the coloured pieces of cloths as previous mentioned. The purpose of this is to create a video especially to the project⁵.

After experimenting with our own footage, we will work on connecting more than one midi / wave output to the extracted data.

The more blocks in the patches, the more processing power we will need. This could be a major problem in the development of a patch that will work proper and without any noticeable delay. One of the solutions to this problem could be networking through several computers. This is the problem to solve in the preceding prototype patch.

1.4 Goals & Expectations

The goal of the project is to obtain a high level of success with the creation of interaction between isolated body movement and sound control/generators. We expect to be able to interpret body movement in a video image and convert the data to generate to sound control. To achieve these goals, we conduct our experiments based on the knowledge obtained in the lectures of Automatic Perception in conjunction with the knowledge gained by working with EyesWeb. At least the first state of the room must be achieved in order to meet our goal.

1.5 Delimitations

In order to achieve a certain amount of success with the outcome of the project, we found it necessary to reevaluate the possibilities of fulfilling our goals according to the original idea. We determined that the creation of the theme room is out of reach and will not take place. We also decided that the sound output for the final patch will be based on midi generated sounds.

⁵ For more details see chapter 4

2 Theoretical background

In this chapter we will describe the theory behind the operation done in EyesWeb.

In the first section we will take a look at projects, which have some similarity to our project. We will do this to uncover the ongoing research.

Second section will be a description of the basic theory used in the project. This knowledge is accumulated from the lectures in Automatic Perception and other resources such as literature, the web and continued discussions.

2.1 State of art

Below follows a description of a few similar projects.

2.2 The EyesWeb project

The main goal of the EyesWeb project is:

“To develop a system able to analyse in real time movement and gesture of one or more humans, with a particular focus to affect or emotion communication”⁶

The approach of the project differs, as they states, from other project by the main issues:

“(i) Our main focus is not only toward symbol recognition from a posture or gesture vocabulary, but toward the extraction of high-level parameters on expressive intentions in the performance.(..)

(ii) With this goal in mind, the analysis techniques we developed are loosely inspired by the work of Laban (1963), and in particular his theory of Movement, called Theory of Effort.(..)

(iii) The interaction metaphors, the requirements, and therefore the system architecture, are considerably different from existing systems;

(iv) We want to extract both two-dimensional and three-dimensional information;

(v) To this aim, EyesWeb typically uses multiple video cameras, but is not based on the classical stereovision approach. A typical on-stage configuration with two cameras consists of frontal and lateral views.”⁷

⁶ Cammuri et.al, 1999, p. 1

Their approach differs from our by the mere fact that they are extracting movements in both the two- and three-dimensional space with two cameras; whereas we only are making our body movement extractions in the two-dimensional space with one video camera for a frontal view of the subject.

2.2.1 SoundBeam™

SoundBeam™ is a “distance-to-voltage-to-midi device, which converts physical movements into sound using information derived from interruptions of a stream of ultrasonic pulses.”⁸

Basically the SoundBeam-technology is based on three major components:

1. A control panel to control which sounds and how sensible the sensors should be to the movements
2. One or more (ultrasound) sensors
3. A computer and amplifier with speakers

The SoundBeam technology is a fairly young product: six years of development has transformed it from “an award-winning concept into a uniquely sensitive, versatile and user-friendly system.”⁹

2.2.2 BigEye

The software BigEye is developed by Studio for Electro-Instrumental Music, Steim. It is “a computer program designed to take real-time video information and convert it into Midi messages”¹⁰. The converting of the video information goes

⁷ ibid, p. 1-2

⁸ <http://www.soundbeam.co.uk>, “home”

⁹ ibid. “About us”

¹⁰ <http://www.steim.org/steim/bigeye.html>

through a colour tracker, which can be programmed to track 16 different objects simultaneous; based on colour, brightness and size¹¹.

Objects and hot zones interact in three distinct ways: a zone sees a new object, a zone sees an old object moving, and a zone sees an object disappear. Each of these three events can generate midi or cause internal program parameters to change.¹²

2.2.3 Palindrome

Palindrome is a German company, which operates with the control of media through human movement and dance. “Our projects of Palindrome range from partnerships with visual artists to design the technical aspects of interactive installations, to a software system for an experimental electro acoustics lab wishing to expand its interactive possibilities”.

We have chosen to describe two of the ways the Palindrome project incorporates the technology: By chest electrodes and transmitter, and by frame grabbing of a dancers movement with a setup of three video cameras.

In the case of the electrodes and transmitter, Palindrome has made the performance “Heartbeat Duet” wherein two dancers wear chest electrodes and transmitters while dancing. The heartbeat of each dancer is converted into a separate musical tone. Supplemental there is a screen behind the dancers which displays a graph of their changing heart rates plotted against time.

The second type of technology is based on frame grabbing, or the capturing of video images in the computers memory. By frame grabbing a dancers movement, and processing them with software, it is essentially possible to convert their movements into music (or other media).¹³

¹¹ Ibid.

¹² Ibid.

¹³ <http://www.palindrome.de/>

2.2.4 Similarities / differences

The EyesWeb based project as well as the other projects can be categorized as programs that incorporate body movement as a controlling parameter.

Common to the projects we have described is, that action by a person causes some reaction in the system. This reaction can be used to control media such as sound and video.

Most similar to our project is the Palindrome project; the differences are the software and the fact that the computer gets a three dimensional view of the dancers on the stage.

2.3 Definition of a digital image

According to Jatila van der Veen, a faculty member at the Department of Physics at the University of California in Santa Barbara, USA, a digital image “is a picture which is divided into a grid of "pixels" (picture elements).”¹⁴

Furthermore each pixel is defined by three numbers (x, y, z) , where the x and y are the coordinates of the pixel's position and the z is the value of the intensity (=colour) of a particular pixel.¹⁵

A colour image is the combination of the three colour channels RGB where the z value represents the sum of the individual channels.

2.3.1 Greyscale images

A greyscale image consists of values ranging from 0 to 255 where 0 is white and 255 are black. When we use the greyscale in conjunction with colour separation, the pixel value of the greyscale image is equal to the pixel value of the separated colour channel value.

¹⁴ der Veen, 2003

¹⁵ Ibid.

2.4 Colour spaces

In short we will describe colour spaces as “a notation by which we can specify colour...”¹⁶.

There are several spaces (HSL: Hue, Saturation and Lightness, HSI: Hue, Saturation and Intensity, HSV: Hue, Saturation and Value, CIE, a colour standard based on brightness, hue, and colourfulness)¹⁷; the colour space we use in the context is the RGB colour space.

2.4.1 The RGB colour space

In the RGB colour space the colours red, green and blue represents the combination of all colours. By mixing the values of the individual pixels we are able to create any colour we need. RGB value 0,0,0 is equal to black and the RGB value 255,255,255 equals white.¹⁸

2.5 Threshold

Threshold can be described as a limitation between a certain value and the actually pixel value. With a threshold we can let a value above the threshold pass and stop values below it.

Our use of threshold is described in chapter 3.

2.6 Digital image processing filters

An image filter performs some operations or a series of operations on an image and returns another image as output.

The digital image processing filters we use in this project can readily be categorized into two groups, one category is non-linear filters and another category is linear filters. One of the differences between those two groups of filters is that the linear filters include and use all the existent data of the image in

¹⁶ Tkališič, p.1

¹⁷ Bourke, 1995

¹⁸ From Volker Krüger's lecture slides: 2. About colours

contrast to the non-linear filter, which do not utilize and preserve all the existing data¹⁹. The non-linear filters are also referred to as convolution filters whereas the linear filter is a rank value filter.

2.6.1 Convolution

“Convolution is an operation, in which a kernel of numbers is multiplied by each pixel and its neighbours in a small region, the results summed, and the result placed in the original pixel location”.²⁰

The discrete version of the convolution is defined as $(f * g)(x) = \sum_m f(m)g(x - m)$
Assume f to be the signal and g to be the filter.²¹

2.6.2 Linear filter

The linear filters in use in our EyesWeb patches are Gaussian filters and high pass filters.

Both types of filters are convolution filters. The key to the difference between a Gaussian and a high pass filter is in the property of the kernel.

The kernel of a Gaussian blur has positive coefficients with weights falling according to a standard distribution away from the centre of the kernel.

A high pass filter will have negative weights at nearby pixels, but a positive weight at the centre.

The Gaussian filter is a smoothing low pass filter and removes the high frequency variability associated with random noise²², conversely the high pass filter that remove the low frequencies and therefore emphasizes the details and edges in the high frequency area. An example of how the two kernels affect the result can be seen in figure 2.1.

¹⁹ The image processing handbook p152

²⁰ The image processing handbook p314

²¹ Computing_with_Images w/ Volker Krueger

²² The image processing handbook p228

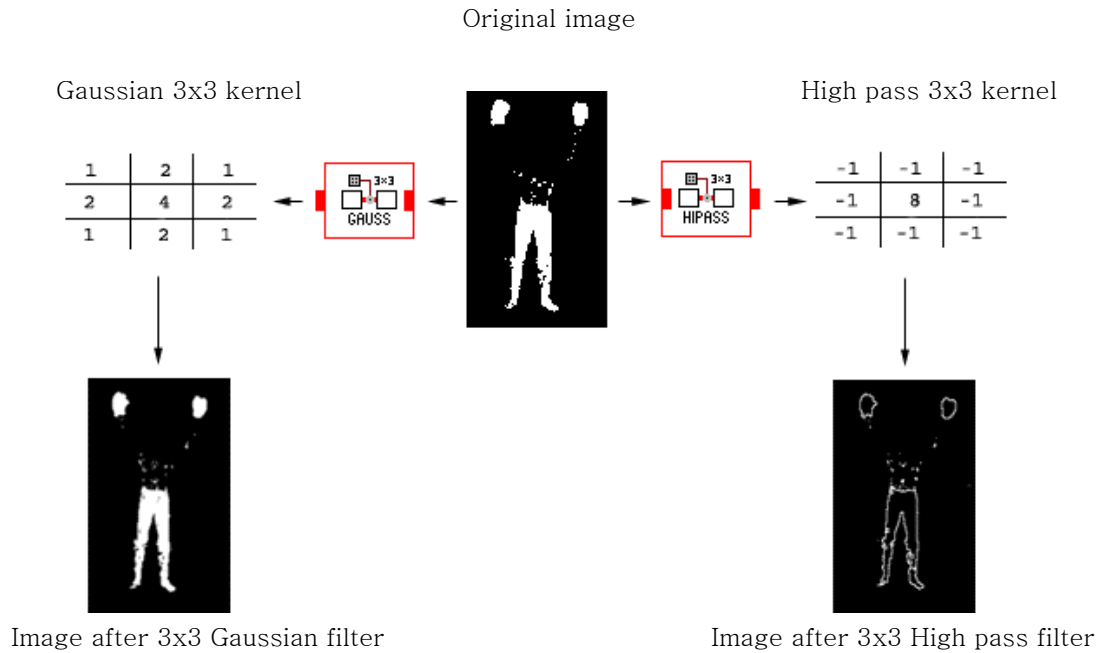


Figure 2.1: Example of two linear filters

2.6.3 Non-linear filter

The median filter is a non-linear filter that is used in one of our preliminary EyesWeb patches. The key word in the median filter is sorting.²³ The pixels within the mask is sorted by ascending grey scale values and the value that is in the middle of the sorted list replace the pixel in the centre of the mask See figure 2.2 below.

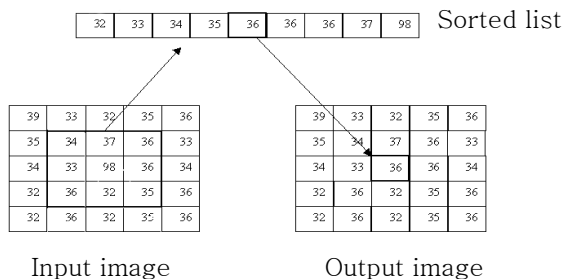


Figure 2.2: Illustration of a median filter 3x3

²³ Digital image processing p123

The median filter is great for removing salt and pepper noise since such noise often occurs like a spike in the pixel value and can therefore easily be removed by the median of the neighbouring pixels. An example of median with a 3x3 mask is shown in figure 2.3.

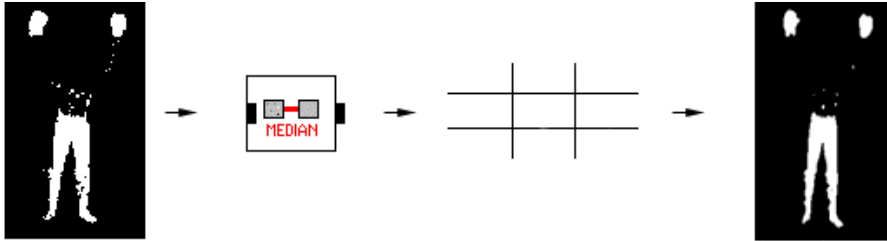


Figure 2.3: Image before and after applying a 3x3 median filter

2.7 Morphological operations

The following paragraph about morphological operators is based on the lectures and slides by Volker Krüger, in the spring of 2003 at the University of Aalborg in Esbjerg.²⁴

Morphological operators are generally used on binary images for the purpose of removing noise in the background or removing holes in foreground/ background.

The operation performed is done by taking a binary image as input and combining the input with a structuring element based on a set of operations depending of the kernel. The structuring element (kernel) can vary in sizes, but usually the values are 0, 1 or none.

The kernel will have its origin in the centre pixel. This is shifted over each pixel of the image and compared with the underlying pixels. If the kernel matches the

²⁴ Volker Krüger: lecture 6: Morphologic Operators, 2003

pixels from the image, the operation is applied to the pixel range in the image. Morphological operation is therefore defined by its structuring element.

2.7.1 Dilate & Close operations

The Dilate operation basically enlarges the foreground and shrinks the background. When the structuring element “touches” the foreground – the value 1 – it will set the input image pixels to be foreground.

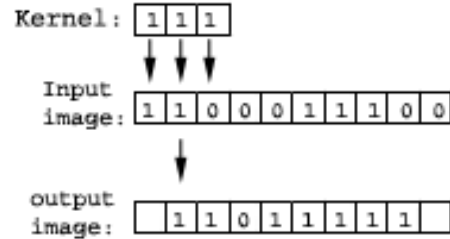


Figure 2.5: Example of dilate operation

The Closing operation is similar to the dilate operation: It shrink background and enlarged foreground; further more it can remove holes.

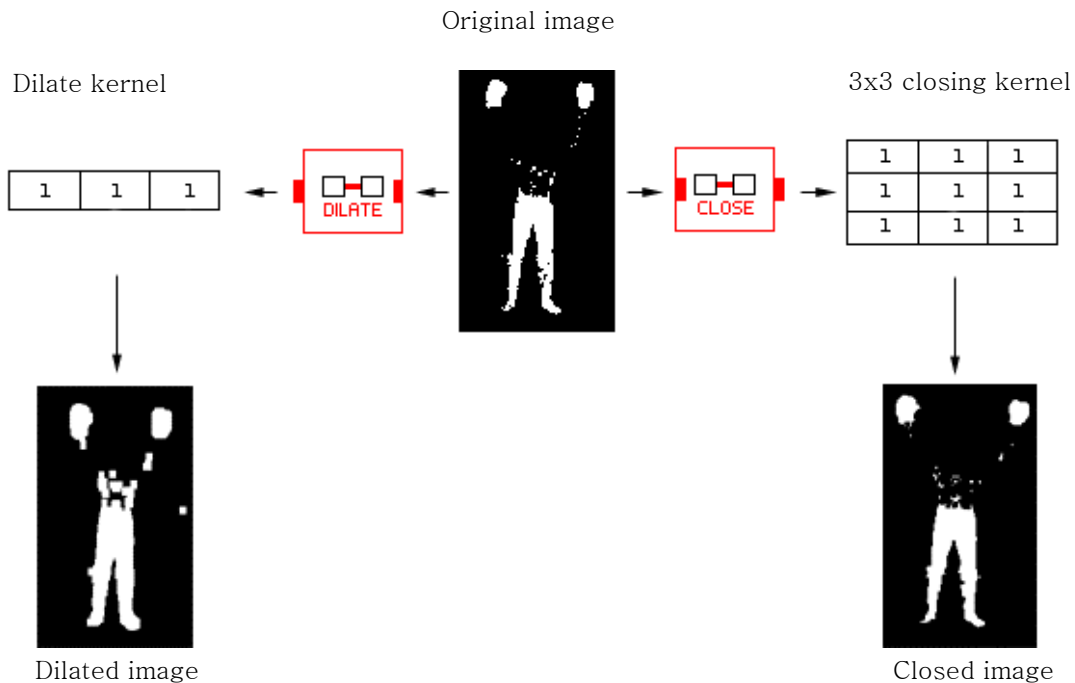


Figure 2.5: Example of dilate and closing filters

As the example in figure 2.5, the foreground has been enlarged and the background is smaller than before the Closing operation; also the holes are closed. The shape looks more uniform now.

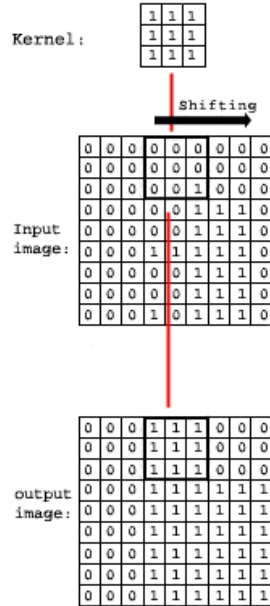


Figure 2.6: A 3x3-structuring element for closing operation

2.8 Logical Operations

The Monadic Operation paragraph and the following about Dyadic Logic Operations will be based on the text “3.7 Logical Operations on Bits” written by Randall Hyde.²⁵

Logical operators are used to determine whether a pixel or data input is within range or out of range of the desired area.

2.8.1 Monadic Operation: NOT

The NOT is a monadic operation; it accepts only one operand, which are:

- NOT 0 = 1
- NOT 1 = 0

²⁵ Hyde 1995-2003

The NOT truth can be illustrated like this:

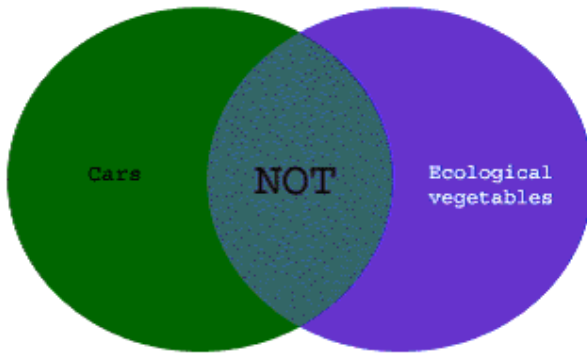


Figure 2.7: The Not operation

We use the NOT operation when we exclude different colours; in the patch “CompleteRedHand” we exclude all other colours than red when separating the red hand.

2.8.2 Dyadic Operation: AND

The AND is a dyadic operation; meaning it accepts two operands which are single binary bits:

- 0 AND 0 = 0
- 0 AND 1 = 0
- 1 AND 0 = 0
- 1 AND 1 = 1



Figure 2.8 The AND operation

The AND operations are in use when we combine the extracted or excluded colours.

3 Experimental setup

This chapter contains the descriptions and the progress of the different patches and blocks that we used in order to develop an acceptable patch according to our goals stated earlier²⁶.

We have divided the task of developing suitable patches in EyesWeb into two separate categories, the audio category and the visual category.

We start by describing three fairly simple audio patches to get a basic idea how audio works in EyesWeb and how we can implement the audio aspect into the visual side of the project.

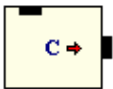
The names of the blocks described in this chapter are named by the developers of the EyesWeb program and are not spelled correctly according to the English language; we have intentionally kept the original names for the sake of consistency with the EyesWeb platform.

3.1 Prototype 1: Midi Test

Blocks used in the EyesWeb patch Midi Test²⁷



Slider control: This block is used for either threshold control or to control the integer value of the generator shown below.



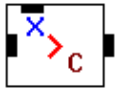
Generator: Outputs various type of signal, for the purpose of our patch the signal type is set to constant.



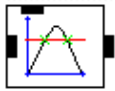
Display: Displays the integer value generated by the generator and slider control above.

²⁶ See Goals & Expectations

²⁷ See figure 3.1: Midi Test



ThreshCondition: Besides prohibiting a signal below the threshold value, it also blocks the input value when not matching a specified threshold condition.



ThreshCrossing: Output the input value every time it crosses (upward, downward or both) a threshold value. In this case the crossing is set to both.



Output: Midi output, send the input value to the specified output midi device, which in this situation is Microsoft GS Wavetable SW Synth for both midi out blocks.

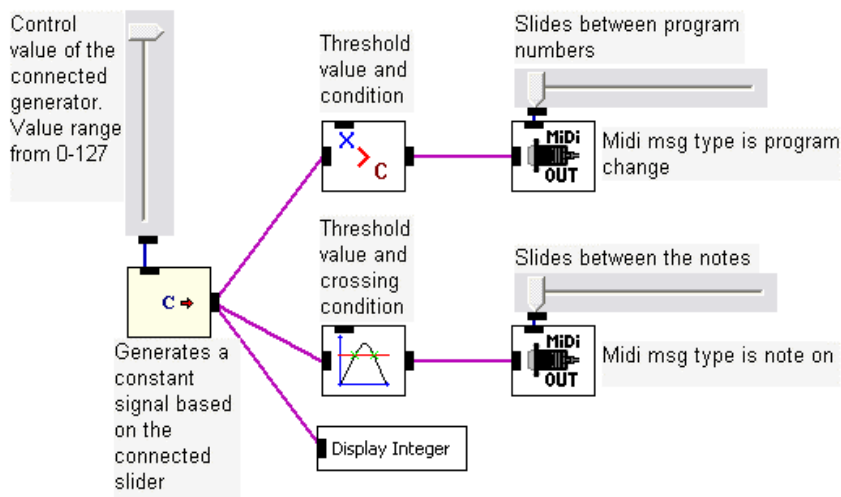


Figure 3.1: Midi Test EyesWeb patch



Miditest.eyw

3.1.1 Objective

This patch is created in order to gain some knowledge of how midi parameters interact and behave in EyesWeb.

3.1.2 Description

The vertical control slider controls the value of the constant signal made by the generator block. The value range between 0–127 and can be seen on the connected display as an integer.

The top threshold block lets signal with an integer value more than 10 pass to the midi out.

The top midi out message type is set to program change, which in our case is the same as change of sounds.

The lower threshold-crossing block in this example has a crossing direction set to both directions and allows signal with a value of 10 or more to generate a note with the corresponding midi out.

The attached slide controller controls what note to play.

3.1.3 Problems

To control the sustain of the note when we generate a note on event with a sound like horns, strings, choir etc. To ensure that the generator only generates one note on event until the user decide to generate another note on event.

3.1.4 Solutions

The threshold-crossing block resolves both problems. First we set a threshold of more than 0, lets make it 10, this creates the effect that a signal below 10 will stop the note on event. Second, the same threshold setting has the ability to control that only one value will pass to the midi out block at a time.

We can add about this patch that the threshold condition block has no effect in this example and can be removed without changing the behaviour of the patch.

3.2 Prototype 2: Wave Test

Blocks used in the EyesWeb patch Wave Test²⁸



Command control: Selected to start; wave file will start when the patch is initialised.



WaveInput: Captures waveform audio from a specified device using standard Windows Multimedia APIs²⁹. Device – allows you to select the input device to be used, among the installed ones. Other parameters such as sample frequency, buffer size, stereo / mono etc can be selected on this block.



WaveFileReaderExternalClock: Determine what wave file to read, the input on this block is used as a clock source.



Control slider: Controls the value of the frequency parameter on the connected low pass block.



XlowPass: Low pass filter that filters out everything below the slider selected frequency value.



WaveOutput: Plays waveform audio on the specified device using standard Windows' Multimedia APIs.

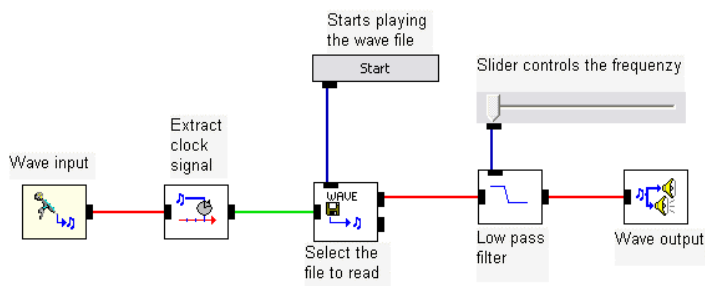


Figure 3.2: Wave Test EyesWeb patch



Wavetest.eyw

²⁸ See figure 3.2: WaveTest

²⁹ Application Programming Interface

3.2.1 Objective

The wave test patch is made in order to explore how wave files can be played in EyesWeb and what kind of manipulation you can apply.

3.2.2 Description

In order to play and hear any wave files, we need to have a wave input where we can change the parameters for our audio device. In this example it is the installed sound card with the selected audio settings like sample frequency set to stereo at 44.1 MHz. We extract a clock signal from the sound stream of the wave input and send it as the clock source to the input of the wave file reader; it is also on this block we choose what wave file to read. The low pass filter filters all the frequencies out below the chosen frequency value by the connected slider. Last we have the wave output block that has to have the same choice of device as the wave input.

The low pass filter serves no other purpose except to illustrate the possibility of manipulating the sound file.

3.2.3 Problem

How to control the start and stop functions of the wave reader?

3.2.4 Solution

Since part of the exported parameters in the wave reader block is start and stop functions and they respond to the data type math, we will be able to use the x and y coordinates of the image to control playback and stop functions of the wave files

3.3 Prototype 3: Tone Test

Blocks used in the EyesWeb patch Tone Test³⁰



Control slider: Controls the value of the frequency parameter on the connected tone generator block.

³⁰ See figure 3.3: ToneTest



Generator: Generates a waveform. The type of waveform may be chosen by means of the parameter ‘Waveform Type’. Possible values are silence, sinusoidal, square or constant, selected type in our example is sinusoidal.



WaveOutput: Plays waveform audio on the specified device using standard Windows' Multimedia APIs.

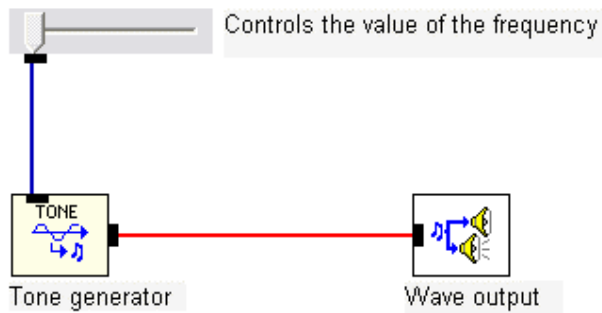


Figure 3.3: Tone Test EyesWeb patch



Tonetest.eyw

3.3.1 Objective

The objective of this patch is to show a third way of generating audio with in the EyesWeb platform.

3.3.2 Description

The generator block generates a waveform, which in our example is the selected type sinusoidal. The attached control slider controls the exported parameter frequency value that determines what frequency to generate. The generator has to be attached to a wave output in order to produce sound.

3.3.3 Problem

How to control the generated frequency as well as start and stop of the generator?

3.3.4 Solution

The exported parameter frequency value on the generator block reads the data type math, which means that we will be able to use the x and y coordinates of a given image to change the frequency. In order to control start and stop of the generator, it can be done with the implementation of one of the threshold blocks.

3.4 Prototype 4: Red Hand

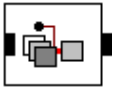
Blocks used in the EyesWeb patch Red Hand³¹



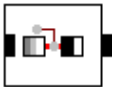
Command control: Selected to start; video clip will start when patch is initialised.



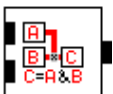
MultimediaFileRead: This block reads a video stream from a selected AVI file.



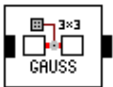
ExtractChannel: Extract a channel from a multichannel image; in this case it is one of the RGB channels.



Threshold: Thresholds the source image. Pixels below the threshold is set to value 0 and pixels that exceed the threshold will get value 255.



DyadicLogicalOp: Perform a dyadic logical operation with the two input images. In this case it is the AND operation.

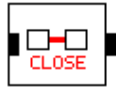


LinearFilterFixed: Perform a linear filtering on the input image; in this case it is a 3x3 Gaussian filter.



Display: Display an IPL image in a popup window. In general a video monitor display.

³¹ See figure 3.4: RedHand



MorphologicalOp: Perform a morphological operation on the input image. The operation here is a close operation that close the holes in the separated red hand

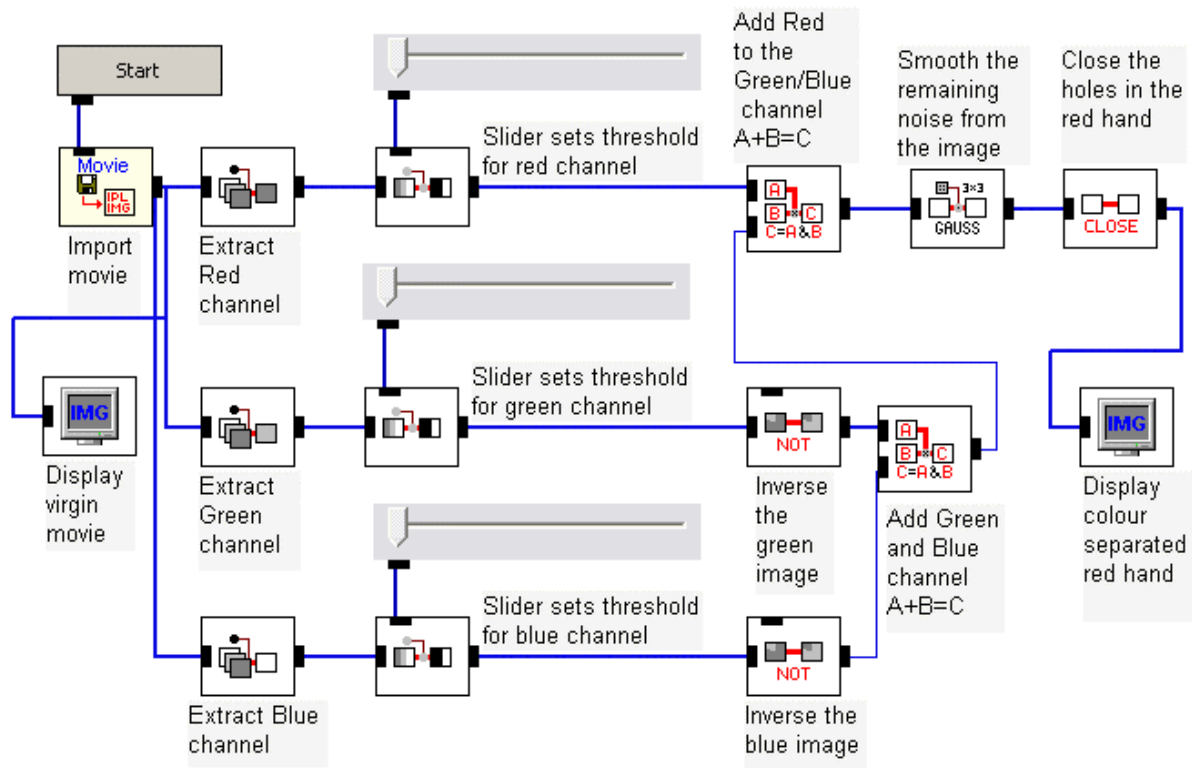


Figure 3.4: Red Hand EyesWeb patch



Redhand.eyw

3.4.1 Objective

The idea of this patch is to isolate and track the red hand of the dancer and be able to get a clear image on the video display

3.4.2 Description

We start by separating the video signal into three colour channels, red green and blue. In EyesWeb the RGB channels correspond to the channels 3, 2 and 1 in the block named extract channel. Then we apply a threshold and adjust the value at each channel. The pixels that exceed the chosen values will have a value at 255 and the pixel that does not exceed the threshold will have the value 0. Since each colour are extracted and represented by a greyscale image, each pixel range in value from 0 to 255 where 0 is white and 255 is black.

After the threshold block the red channel is ready for the Dyadic Logical Op block.

On both the green and blue channels we insert a block that performs monadic logical operations. In this case the block is set to a not operation and invert the green as well as the blue channel. Next we perform the logical operation AND, between the green and the blue channels. The resulting channel will continue to a similar block that carries out the AND operation with the red channel. We are now back to one channel which we send through a Gaussian 3x3 smoothing filter in order attempt to smooth away the remaining pixels that we don't want and as a preparation to the morphological operation close, which closes the "holes" and create a more visible image of the red hand.

3.4.3 Problem

One of the problems in creating this patch has clearly been to isolate the red hand from the almost as red floor. To remove the colours of the image and at the same time is able to distinguish the red hand from the rest of the image.

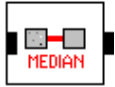
3.4.4 Solution

When addressing the problem above, we focus on fine tuning the threshold setting of the green and blue channels before inverting those channels, which in reality is the same as removing as much of those colours as possible and still leave room for the red colour to appear when added together. When red channel threshold is set

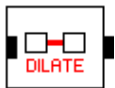
to 100, added to the other channels and afterwards smoothed by a Gaussian filter, there is basically nothing left but the red hand.

3.5 Prototype 5: Complete Red Hand

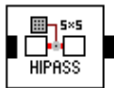
Additional blocks used in the EyesWeb patch Complete Red Hand³²



NonlinearFilter: Perform a nonlinear filtering on the input image. Here it is a median filter to eliminate noise on the input image.



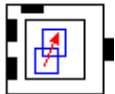
MorphologicalOp: Perform a morphological operation on the input image. The operation here is a dilate operation.



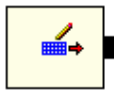
LinearFilterFixed: Perform a linear filtering on the input image; in this case it is a 5x5 High pass filter.



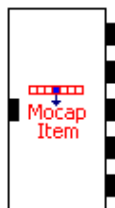
ExtractRegions: Identify multiple regions, calculating their bounding rectangle approximation.



TrackRegions: Track and identify multiple regions extracted by the Extract Regions block. In this case we have one region to track and identify.



Generator: Generates a matrix based on the data provided by the Extract Regions block



Extract: Extract an item from input MoCap list, in this case it is the input matrix created with the data extracted by Extract Regions and Track Regions.

³² See figure 3.5: CompleteRedHand



GetEntry: Get the entry from the input matrix and create the output MathDatatype.Scalar.



Display: Displays the input scalar value. In this case the value displayed is representative of the pixel row that equals the placement of the red hand on the x-axis.



Generator: Generate a waveform. Possible values are Silence, Sinusoidal, Square or Constant. In this case the block is set to sinusoidal and the exported parameter is set to frequency. The frequency change according to the entry extracted from the Input matrix displayed in Display_1.



WaveOutput: Play waveform audio on the specified device using standard Windows' Multimedia APIs.

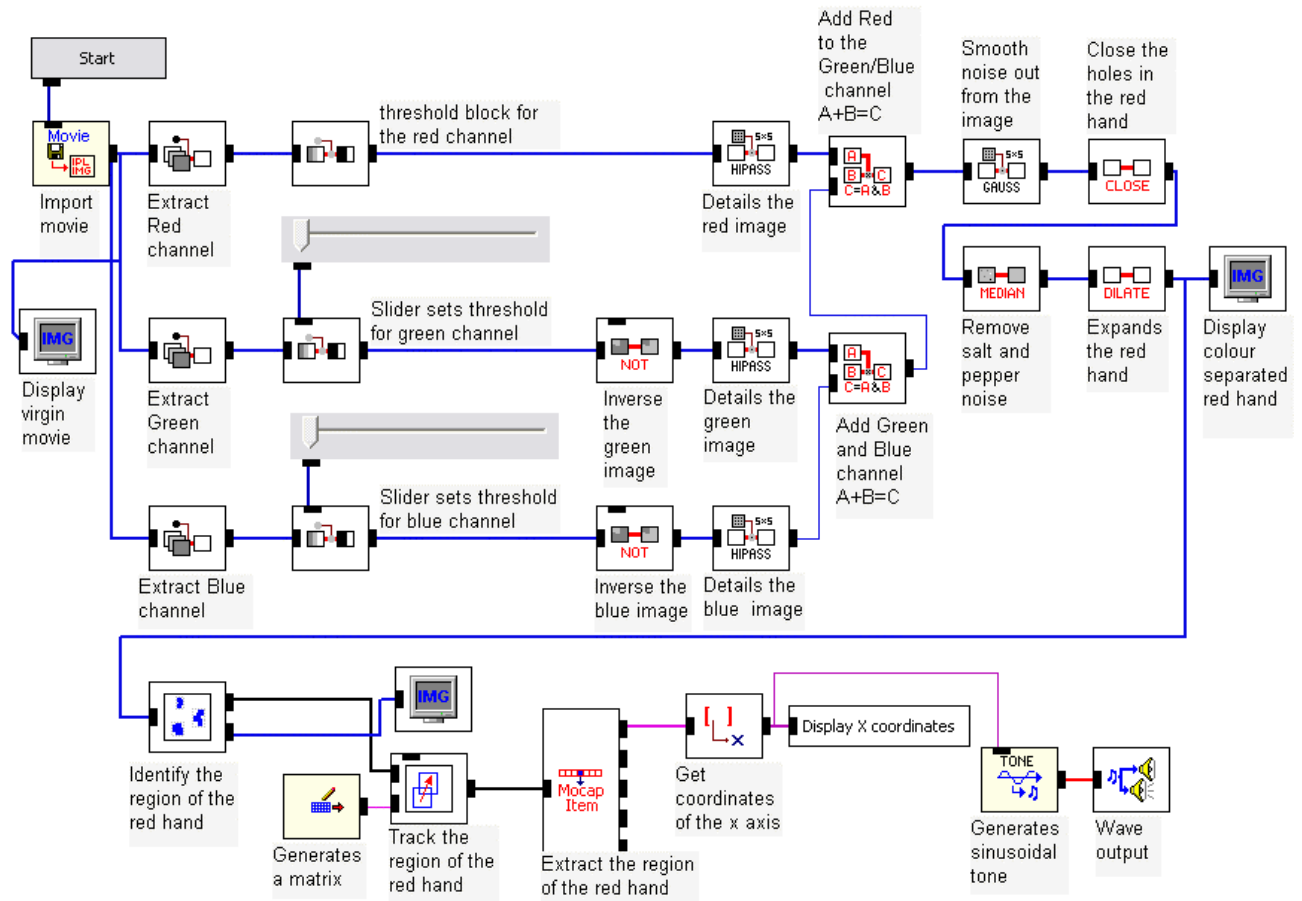


Figure 3.5: Complete Red Hand EyesWeb patch



CompleteRedHand.eyw

3.5.1 Objective

The object of this patch is to extract the coordinates of the red hand, so that we in this example can generate different sinusoidal frequencies in conjunction with the horizontal movement of the red hand. In this case the x-axis of the video image is equal to the potential frequency range of the tone generator. It is the actual movement of the hand on the x-axis, when the hand is visible, that will determine the frequency of the tone. When the hand is invisible the last known frequency will continue to be heard until the hand appears visible again and a new frequency will be heard.

3.5.2 Description

In this version of the patch we added several blocks compared to the version described as prototype 4. We also added a complete track region section with a sinusoidal tone generator. We start with the added filters and the morphological operators dilate. We inserted, after the inverted red, green, and blue channels, 5x5 high pass filters. We do this in order to sharpening the images before we twice perform the dyadic logical and operation that combine the separated channels back to one channel. The combined channel continues to a 5x5 Gaussian filter that blurs the remaining pixels for better region tracking. Next step is the rather large 9x9 median filter to remove the last salt and pepper noise before we choose to dilate the remaining part of the image in order to expand the size of the separated red hand.

In order to extract the coordinates of the red hand we added a whole section of blocks whose primary function is to identify, track and get the coordinates of the red hand.

First block is the Extract Regions that identify the region containing the red hand, calculating the bounding rectangle approximation region. The next block is a track region block that tracks the coordinates of the previous identified region. This block is connected to a matrix generator that generates a matrix based on the tracking information of the track region block. The Mocap extraction block extract items from the input motion capture list, which is the matrix information gathered from the track region block. The matrix information continues to the get entry block, which gets the information as the data type math and send it as integers to both a display and to the frequency parameter of the tone generator. Thus the numbers you see on the display x coordinates is equal to the frequencies generated by the sinusoidal tone generator. We have tested this with a frequency analysis block.

3.5.3 Problem

The massive median filter (9x9) slows down the video considerably and so does the section with the blocks that extracts the image coordinates.

The path of the red hand is only really visible on the video display, when viewing the display immediate connected to the block Blob Analysis Extract Regions.

3.5.4 Solution

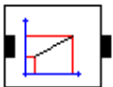
In order to create a patch that can work in a real time digital video environment, we will have to come up with a patch that has a more direct approach to colour separating, a patch that does not need the massive filters you just experienced. The first step to achieve this is to shoot our own video in a cleaner environment than the micro-dance video.

3.6 Prototype 6: Choice Project

Additional blocks used in the EyesWeb patch Choice Project³³



Baricenter: Extract the coordinates of baricenter from the input Image. In our patch this block refers to the baricenter of the colour separated hands.



LinearRescale: Rescale the Input Value. In our example it rescales the width of the x-axis from 320 to the midi standard of 127.

³³ See figure 3.6: ChoiceProjectPatch

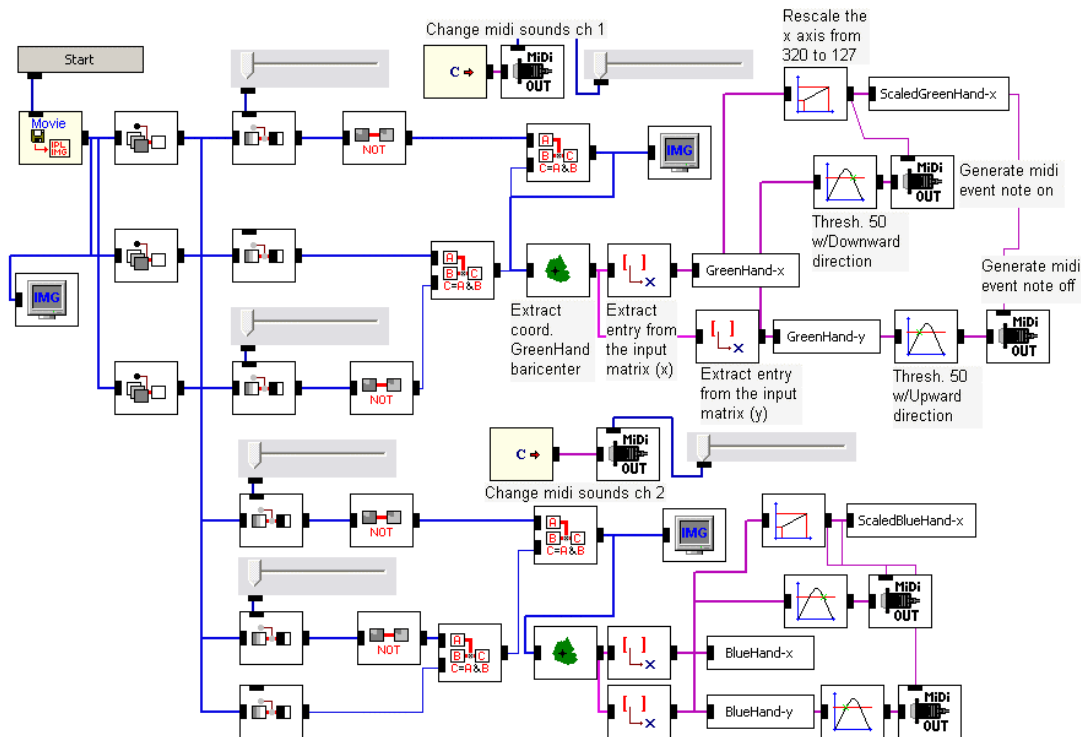


Figure 3.6: Choice Project EyesWeb Patch (Only relevant new blocks is commented)



ChoiceProject.eyw

3.6.1 Objective

The objective of this patch is to create a patch that works with two colour separations, different midi sounds for each separated colour and the ability to work with video footage as well as a real time DV camera set up.

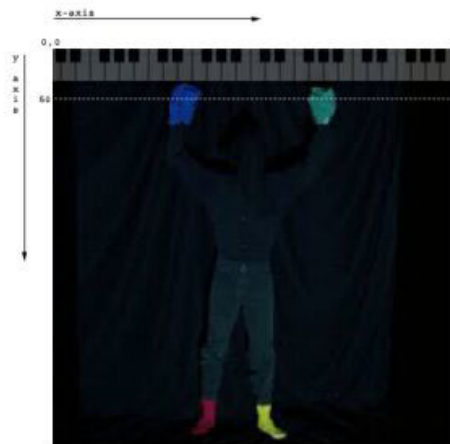
In essence we want to be able to play a virtual keyboard when we raise our hands in the air.

3.6.2 Description:

The colour separation of this patch is based on the same techniques as the previous patches and therefore already thorough explained. The only difference here is that the video has been shot in a controlled environment thus the need of filters literally has been eliminated and any delay between the video and sound interaction is non existent due to the fact that the only source of sound is Microsoft GS Wavetable SW Synth.

We separate the two different coloured hands one at a time, after each separation we apply a baricenter block to each channel. The object of the baricenter block is to extract the coordinates of the baricenter within the image, i.e. the separated green or blue hand. The get entry block extracts an entry from the input matrix, which in this patch is derived from the baricenter block. The get entry block has parameter settings that enable you to choose between rows or columns, this is the same as x or y coordinates. We use both sets of coordinates where the x-axis determines which note will be played and the y-axis determines when the same note will be played on the midi out block.

The control of the virtual midi keyboard, as shown in figure 3.7, is achieved with the combined use of the threshold blocks and the scalar block. The scalar block has this great feature that it can rescale the original 320-pixel width of the image down 127 units, measuring the same width. Since 127 is the numeric value used in various midi parameters, it made sense to apply this value here.



to

Figure 3.7: Virtual keyboard

By rescaling the x-axis to 127 units and connect to control on the exported parameters of the midi out block, it is in reality the position of the coloured hands on the x-axis that determines what note to play.

The threshold crossing block with a value of 50 and downwards crossing direction for the note on event determines when to play the note. It is illustrated in figure 8 by the dotted white line on top of the picture. Since pixel 0, 0 is the top left corner of the image you have to cross the line in an upward motion in order to trigger the note on event.

The note off event has also threshold value 50 but an upward crossing direction which makes it opposite to the note on event. The midi out blocks connected to the

green hand is generating notes on midi channel 1 and the midi blocks connected to the blue hand are set to midi channel 2. This explains why the two hands play different sounds.

3.6.3 Problems

It is not possible at the moment to play more than one note at a time for each hand. There is still a problem with certain sounds and the after touch, when sounds like choir, strings etc. is chosen and played, the notes just keep playing.

3.7 Patch conclusion

This concludes our chapter of EyesWeb patches and how we by starting with the simple patches and solved one problem at a time were able to put together a working patch which is acceptable according to our goals stated earlier in this report.

4 The Experiments

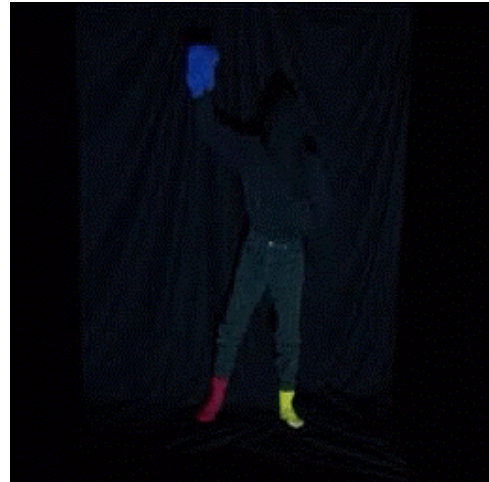
In this chapter we describe the room, the video shoot used prior to the real time experiments and the real time tests.

4.1 The setup of the room

The idea of the room has from the start been that it should be mobile/portable: We should be able to implement the room in places like the mall, exhibition or museum halls, in the street etc. The only thing that is needed in the setup is a black theatre backdrop and different coloured gloves and socks.

4.1.1 Description

Our version of the room is a makeshift room with black back drops on three of the walls and the floor is equally covered by a black curtain, as seen in figure 4.1. We are using the black curtains to ensure a “steady” background and to control the light for the sake of the later colour separation. For The real time experiments we will make use of



two halogen spots on stands, placed in front and off to the sides of the subject.

Figure 4.1: The room

4.2 Preliminary video footage

In order to move away from the massive amount of image processing in the red hand EyesWeb patches, we decided to shoot some new video footage with focus on colour separation. For the sake of the later colour separation, the subject in the video was dressed in black except some very colourful clothing on hands and feet.

This set up in conjunction with the background and the high-powered light source, a LCD projector with a 1000-LUX, resulted in some very bright coloured video.³⁴

The first attempt did not have the expected effect in our patch and created a substantial delay between video and audio, this due to the size of the video file caused by length of video and to high an image resolution.

After the second attempt we had three short and usable video clips in a resolution of 320 x 256 pixels and with duration between 12 and 17 seconds.

The new footage paved the way for a new set of EyesWeb patches, which did not need as much processing power as the previous patches.

4.3 Real time test

In order to test the patch in real time we need to modify the choice patch for it to work with the DV camera. We do this by replacing the MultimediaFileRead block with the DVCamInput block. In addition to this, the priority parameter on the DV block must be set to idle to in order to avoid critical delay on the video stream.

The DV patch proved almost as effective as the choice patch in regards of separating the colours and extract the coordinates of the moving hands. A slight delay in the computation of the live video stream compared to the preliminary video footage caused an acceptable delay, approximate 300 – 400 milliseconds, between the live action and the corresponding sound output of the computer.

One reason of the delay is that the standard resolution of the DV camera is 720x576 compared to our video, which is half the resolution.

At the moment we have no possibility of changing the resolution on the DV camera and therefore accept the delay and find that the real time test works well with the goals and expectations stated earlier³⁵.

³⁴ *“Light unit for measuring illumination. It is defined as the illumination of a surface when luminous flux of 1 lumen falls on an area of 1 m². It is also known as lumen per square meter, or meter-candelas”*
(From: <http://www.cctvlabs.com/Theory/l.html>)

5 Summary and discussions

In this chapter we summon the results that we achieved during the development of the prototype patches and the experiments with these.

Furthermore we will discuss how the results can be related to other projects.

We will explain our vision with further development of the project and the usage of an end product based on the project.

5.1 Problems during the experiments

During this project we have encountered a range of different problems. We categorize the problems as: preliminary video and real-time experiments.

5.1.1 Preliminary video recordings

First we used the video footage “micro-dance.avi” to test the colour separation in the red hand patch. For this purpose the video proved efficient, although the “noise” produced by the shades of red in the floor created obstacles in the later versions of the red hand patches.

The solution to this problem demanded the use of several large filters in order to remove the “noise”. This unfortunately slowed down the patch to a non-acceptable level. As a way to optimise the patch we decided to shoot some new video footage. These video recordings proved sufficient for the rest of the preliminary patch experiments in EyesWeb.

5.1.2 Real-time experiments

To make the real-time experiments feel as realistic as possible the delay between action and sound has to be minimal. During the first experiment in real-time the delay reached as much as 2 seconds, after several adjustments we decreased the delay to approximate 300-400 milliseconds. The delay is an acceptable problem at

³⁵ See chapter 1.5 Goals & expectations

the moment but require further work in order to bring it down below 100 milliseconds and eventually no delay at all.

5.2 Did we achieve our goals

In this paragraph we will determine if our goals have been reached.

5.2.1 Transformation of movement into sound

We achieved this goal when a person, with the appropriate gloves, was able to trigger notes on the virtual keyboard and generate sound.

5.2.2 The theme room

Our goal with the theme room was very modest. We had a vague idea of creating a room filled with sound. We achieved to play simple sounds based on movement of the person. This is the foundation in creating a theme room.

5.2.3 Interaction

This gave the users the sensation of controlling the sounds with their body movements. Furthermore the users found it interesting and fun that different spots in the air created different sounds. Unfortunately no new Mozart was found among the users.

5.3 Real-time body synthesizer vs. Recorded footage

Recorded footage can be optimised to run smoothly and with out delay, real-time on the other hand have a fixed video resolution which demand more computing power and therefore creates a slight delay as previous mentioned. A possible solution to this problem might be combining the processor power of several computers in a network; again this will take some more research and experimenting in order to solve this problem.

5.4 Video camera control vs. Ultra-sound (SoundBeam™)

The two solutions for converting movement into midi messages have their advantage and disadvantage.

The advantage of the ultra-sound solution is the way you actually play the beam it feels like you play an instrument.

Because of limitations in the sensor range the users interact within a defined area that seems less flexible than the room created with a DV camera.

In the area of midi control, SoundBeam has a lot of experience that makes comparison impossible.

5.5 Visions with the further development

As described in the chapter “Introduction” our initial idea was to create a theme room where the user would choose a theme and create a soundscape related to the users choice. We did not create the theme room but feel that the initial idea is realisable and that it is within our reach.

5.6 Visions with the usage of the end-product

We see possibilities for the end product in a lot places and situations: In the mall, in exhibition and concert halls, educational and other institutions, in therapeutic as well as rehabilitation sessions and so forth. In our vision the system can be used by anyone: Children, adults, elderly, people with special needs, people from everywhere etc.

6 Conclusion

Our main goal with this project was to create a room, where different body movements resulted in various midi-generated sounds for either internal or external playback.

We achieved our goal by implementing the theory obtained from the lectures by Volker Krüger at Aalborg University Esbjerg.

The result of the project “Body Synthesizer” is a patch constructed in EyesWeb by creating, combining and experimenting with the different blocks and patches.

The “Body Synthesizer” makes it possible for a user to trigger sounds based on the users body movement. ‘Play with your body’ so to speak.

We consider “The Body Synthesizer” a good foundation for future projects and believe that there are a lot of possibilities for expansions within the concept.

7 Literature/Resources

Antonio Cammuri, Matteo Ricchetti & Ricardo Trocca : “ EyesWeb – Toward Gesture and Affect Recognition in Dance/Music System”, Laboratorio DIST di Informatica Musicale, Università di Genova (<http://musart.dist.unige.it>) , Genova, Italy, 1999

Paul Bourke: “RGB colour space – Lists of RGB values for named colours”, online: <http://astronomy.swin.edu.au/~pbourke/colour/>, Swinburne University of Technology, Australia, 1995

EyesWeb : online : <http://infomus.dist.unige.it/eywindex.html>, Laboratorio DIST di Informatica Musicale, Università di Genova (<http://musart.dist.unige.it>) , Genoa, Italy, [udgivelsesår]

Randy Hyde: “The Art of Assembly Language”,

online: http://webster.cs.ucr.edu/Page_AoAWin/HTML/AoATOC.html, Department of Computer Science & Engineering, University of California Riverside, USA, 1995–2003

Bernd Jähne: “Digital Image Processing”, 5th revised extended edition, Springer – Verlag Berlin – Heidelberg New York, 2002

Volker Krüger: Automatic Perception (lecture 1–8); lecture series at University of Aalborg Esbjerg, Denmark, online: <http://bokertov.cs.aue.auc.dk/~vok/>, spring term 2003

Palindrome – online: <http://www.palindrome.de/>

John C. Russ: “The Image Processing Handbook”, Fourth edition, Materials Science and Engineering Department, North Carolina State University, Raleigh, North Carolina, USA, CRC Press, 2002

SonudBeam™ – online: <http://www.soundbeam.co.uk/>

Steim – online: <http://www.steim.org/steim/bigeye.html>

Marko Tkaličič: “Colour spaces – Project for the Digital signal processing course”,
online: http://ldos.fe.unilj.si/docs/documents/20021113135017_markot.pdf, Faculty of electrical engineering, University of Ljubljana, Slovenia,

Jatila van der Veen: “What exactly is a digital image?”, online:
http://www.physics.ucsb.edu/~jatila/astro/astro2/ccdlab_intro/tsld003.htm,
University of California, USA, 2003