A speech driven video texture talking head
J. B. Moxon
BSc Computer Science with Economics (Ind)
(2004/2005)

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) _______________________________
Summary

This project involves the conception and implementation of a speech driven video texture talking head. In doing so it deals with the notion of video textures and the techniques involved in producing them.
Acknowledgments

I would like to express my gratitude to my project supervisor Dr Chris Needham for his time, knowledge and insights. For his frequent chasing up of setup problems and software licence issues. Most of all for the numerous scraps of paper and scribbled diagrams, which were invaluable in aiding my understanding.
Contents

1. Introduction
   1.1. Report outline .......... 01
   1.2. Aims and objectives .......... 01
   1.3. Minimum requirements .......... 02
   1.4. Possible enhancement .......... 02
   1.5. Deliverables .......... 03
   1.6. Schedule .......... 03
      1.6.1. Revised schedule .......... 04

2. Background Research
   2.1. Established techniques and their limitations .......... 06
   2.2. The concept of video textures .......... 06
      2.2.1. Methodology .......... 06
      2.2.2. Conceptual stages .......... 07
   2.3. Characteristics desirable to video texturing .......... 07
   2.4. Advanced video textures .......... 08
   2.5. Generalised media textures .......... 08
   2.6. Driving video textures .......... 09
   2.7. Closing Discussion .......... 09

3. Analysis and design
   3.1. Application outline .......... 10
      3.1.1. Training .......... 10
      3.1.2. Execution .......... 11
   3.2. Approach and methodology .......... 12
   3.3. Tools and applications .......... 13
4. Implementation
   4.1. Initial experimentation ........................................... 14
   4.2. Transition to Matlab ............................................ 15
   4.3. Analysis and training ........................................... 16
      4.3.1. Image processing and representation ...................... 16
      4.3.2. Audio processing and representation ..................... 18
      4.3.3. Image and audio clustering ............................... 20
   4.4. Input audio analysis ........................................... 25
   4.5. Video generation ................................................ 25
      4.5.1. Cluster choice ............................................. 25
      4.5.2. Frame choice ............................................. 26
   4.6. Video Construction ............................................. 26
   4.7. Enhancements ................................................... 26
      4.7.1. Improved frame choice .................................... 26
      4.7.2. Video cross-fading ....................................... 28
      4.7.3. Two way cross-fading .................................... 28

5. Evaluation
   5.1. Prototype appraisal ............................................ 29
      5.1.1. Audio synchronisation .................................... 30
      5.1.2. Articulation .............................................. 32
      5.1.3. Frame choice and articulation ........................... 34
      5.1.4. Video dynamics ........................................... 34
   5.2. Conclusion ..................................................... 38
   5.3. Extensions ..................................................... 38

References ............................................................. 40

Appendices
   Appendix A - Reflection
   Appendix B – Operation Instruction
Chapter 1

Introduction

1.1. Report outline

The first chapter of this report addresses the foundation of the project. It lays out the initial motivation behind the project as well as what should be achieved. Chapter 2 provides an overview of the work from which the project draws. The theoretical basis, methods and concepts involved are examined and relevant subject areas are addressed.

Chapter 3 presents an outline for the application development process. The overall layout of the application is detailed and its’ fundamental components are reviewed. This section also indicates the approach for development and discusses suitable tools for implementation. Chapter 4 moves to the production of the solution. Each element of the application is presented and major design and implementation decisions are examined. The theoretical aspect behind these decisions is also presented and justified using relevant sources.

Chapter 5 concludes by evaluating the prototype application and project as a whole. It examines achievements related to minimum requirements (1.3), inspects the output produced as well as providing an appraisal of overall results.

1.2. Aims and objectives

The overall aim of this project is to produce a speech driven video texture talking head. This would ideally be used for admissions day demonstration.

This aim was essentially derived from a number of general objectives: Admissions day can be though of a showcase for the school of computing, the teaching and research it conducts. With this in mind exhibits and activities during the day should be stimulating as well as informative. Any application or demonstration used in the event should engage visitors as well as provide an insight into the work of the school.
In considering the above objectives, a speech driven video texture talking head would:

- Offer an interactive exhibit.
- Be visually simulating and engaging.
- Provide human computer interaction with means other than a mouse and keyboard.
- Demonstrate advances and new developments in computing.
- Show new areas of computing that are not yet considered mainstream.

### 1.3. Minimum Requirements

1. Prototype software that creates some form of synthetic video in response to an audio signal.

2. Demonstrate the above software with a single user on a predetermined sentence or limited vocabulary of sound.

### 1.4. Possible Enhancements

In addition to the minimum requirements indicated above it will possible to extend the project. Areas of enhancement might include:

1. General improvement of the methods used within the application. More advanced means of input analysis will be achievable. Advancements in the representation and construction of the final video texture can also be considered.

2. Acceptance of a larger range of audio input. Rather than the predetermined sentence or vocabulary indicated, the software could be extended to allow a more extensive set vocalisations. The processing of free speech could be considered the ultimate, but perhaps unrealistic, aim.

3. The application could be advanced to allow speech input from different users.

4. Improvements could be made to the quality of visuals. It is likely that enhancements in image quality could be implemented, or an increase in complexity of the visual modelling techniques used.
5. The software could be advanced to provide a variety of faces. This might include collecting input video and constructing textures in a short space of time.

6. Provision could be made for continued use (or advancement) of the application. Items such as user guides and setup documentation could be produced.

1.5. Deliverables

With minimum requirements in mind, the deliverables are:

1. Software for the capture and processing of video and audio input. This is used in the training and setup of the second deliverable.

2. Software that uses the models and results gained from the training data. It will control, visualise and operate the talking head.

The deliverables applicable to the extensions mentioned above include:

1. User guide documentation on the setup and operation of the software detailed above
   (Provided in appendix B)

1.6. Schedule

<table>
<thead>
<tr>
<th>Completion Date</th>
<th>Task/Event/Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>14th October</td>
<td>- Initial meeting with project supervisor</td>
</tr>
<tr>
<td>21st October</td>
<td>- Complete initial reading on video textures and techniques.</td>
</tr>
<tr>
<td></td>
<td>- Decision made on project topic/area (in line with general objectives)</td>
</tr>
<tr>
<td>28th October</td>
<td>- Arrange loan of webcam</td>
</tr>
<tr>
<td></td>
<td>- Finish examination of previous work</td>
</tr>
<tr>
<td></td>
<td>- Produce draft aims and minimum requirements for review</td>
</tr>
<tr>
<td>29th October</td>
<td>- Submit aims and minimum requirements</td>
</tr>
<tr>
<td>4th November</td>
<td>- Complete initial work on image and audio capture and conversion.</td>
</tr>
<tr>
<td></td>
<td>- Review sample applications and examine Real-Time Image Library</td>
</tr>
<tr>
<td>11th November</td>
<td>- Produce sample threshold application with Real-Time Image Library</td>
</tr>
</tbody>
</table>
- Capture sample video of a variety of faces

**18th November** - Consider methods of image processing/representation

**26th November** - Plan mid-project report for review
- Consider methods of audio processing/representation

**3rd December** - Complete mid-project report draft for review
- Summarise conceptual architecture for application

**9th December** - Complete mid-project report second draft for review

**10th December** - Submit mid-project report

**24th January** - Mid-project report returned, review and refine

**1st February** - Complete input and analysis stage of application

**10th February** - Complete system using basic analysis and methods. From this point onwards models and methods should be improved and refined

**1st March** - Testing and refinement of software
- Draft table of content and draft chapter for review

**10th March** - Second draft of table of content and draft chapter for review

**11th March** - Submit table of content and draft chapter

**18th March** - Deadline for completion of progress meeting

**7th April** - Conduct software demonstration

**14th April** - Draft project report for review

**21st April** - Second draft of project report for review

**27th April** - Submit project report

### 1.6.1. Revised Schedule

Due to changes in implementation techniques and setbacks with software setup, minor modifications to the initial project schedule were necessary. The revised schedule is shown below:

<table>
<thead>
<tr>
<th>Completion Date</th>
<th>Task/Event/Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>14th October</td>
<td>Initial meeting with project supervisor</td>
</tr>
<tr>
<td>21st October</td>
<td>Complete initial reading on video textures and techniques.</td>
</tr>
<tr>
<td></td>
<td>- Decision made on project topic/area (in line with general objectives)</td>
</tr>
<tr>
<td>28th October</td>
<td>Arrange loan of webcam</td>
</tr>
<tr>
<td></td>
<td>- Finish examination of previous work</td>
</tr>
<tr>
<td></td>
<td>- Produce draft aims and minimum requirements for review</td>
</tr>
<tr>
<td>29th October</td>
<td>Submit aims and minimum requirements</td>
</tr>
<tr>
<td>4th November</td>
<td>Complete initial work on image and audio capture and conversion.</td>
</tr>
</tbody>
</table>
- Review sample applications and examine Real-Time Image Library

**11th November** - Produce sample threshold application with Real-Time Image Library
- Capture sample video of a variety of faces

**18th November** - Consider methods of image processing/representation

**26th November** - Plan mid-project report for review
- Conclude Matlab sound exercise

**3rd December** - Complete mid-project report draft for review
- Summarise conceptual architecture for application

**9th December** - Complete mid-project report second draft for review

**10th December** - Submit mid-project report

**24th January** - Mid-project report returned, review and refine.
- Transition of existing C++ work into Matlab

**3rd February** - Complete input and analysis stage of application

**10th February** - Complete system using basic analysis and methods. From this point onwards models and methods should be improved and refined

**3rd March** - Testing and refinement of software
- Draft table of content and draft chapter for review

**10th March** - Second draft of table of content and draft chapter for review

**11th March** - Submit table of content and draft chapter

**18th March** - Deadline for completion of progress meeting

**7th April** - Conduct software demonstration
- Implement Extensions

**14th April** - Draft project report for review

**21st April** - Second draft of project report for review

**27th April** - Submit project report
Chapter 2
Background Research

2.1. Established techniques and their limitations

Current computer graphics rendering techniques are constrained by the inability to produce truly photorealistic results. Motion capture, texture mapping and various artificial intelligence techniques provide superb results rendering rich and lifelike environments. However, the intrinsic ability of the human visual and cognitive system to perceive even tiny discrepancies or inaccuracies means we are often not fooled. Rather than recreating scenes, techniques can focus on allowing us to draw from the natural environment and rearrange it to suit our requirements. One such medium in this field is video textures.

2.2. The concept of video textures

Video textures can be defined as “a continuous, infinitely varying stream of video images”. This concept involves the ability to use traditional video clips, analyse and rearrange them to recreate distinct video that maintains the essence of the original. These textures can theoretically be of arbitrary length and maintain the content and dynamic of the original media. They therefore provide the opportunity for the introduction of active photorealism into a scene (Schödl et al, 2000, p489).

2.2.1. Methodology

Schödl et al provide an intuitive overview of the process of constructing video textures. The technique begins with a segment of input or “training video”. This is broken down into smaller segments, usually frames or small sections of video. Theses elements are analysed in order to find ‘good transition points’ between them (Schödl et al, 2000, p490).

With the “video structure” defined two avenues are presented for recombination of the elements. The first is ‘random play’. This involves frames being sequenced in a deterministic manner. The previous frame and knowledge of appropriate transitions from it allow frames to be placed in a suitable order.
Alternatively a loop approach can be used. Here a certain number of elements/frames are chosen. After the passing of this number of elements the sequence created should have returned to its start state i.e. looped. Using this foundation the appropriate frames are selected based on suitable transitions and the desired start (and end) point. Regardless of the method used to determine the order of frames, the final stage is to combine the chosen sequence to create the video texture with a suitable “visually pleasing” result (Schödl et al, 2000, p490).

2.2.2. Conceptual stages

Phillips and Watson further analyse the video texturing process splitting it into three conceptual stages. These are “analysis, synthesis and rendering of the final video texture”. During the analysis stage we derive suitable transitions between frames based on the differences between them. Steps must be taken to address and preserve the dynamic of the input. The main example used to illustrate this is the case of a swinging pendulum. Consideration must also be given to transitions from which the there are no feasible continuations or “dead ends”. Synthesis is the reordering of the frames, be it by the random play or looping approaches indicated above. Finally rendering recombines the frames, possibly using techniques such as blending, cross-fading or morphing to avoid obvious transitions between frames (Phillips and Watson, 2003, pp 9 - 10).

2.3. Characteristics desirable to video texturing

Previous work examines various subjects in the context of video textures. The varying results experienced can generally be attributed to the make up of the training sequence feeding the texture. Consensus suggests that facial images proved a sound foundation for producing video textures (Schödl et al, 2000; Phillips and Watson, 2003). These wide ranging investigations appear to be supported by more advanced face modelling techniques and attempts to synthesise speech visuals (Bregler, Covell and Slaney, 1997; Cosker et al, 2004).

The nature of movement and dynamics of the face mean it works well as an individual texture. It is, however, difficult to underline the characteristics that produce a good video texture. One aspect seems to be the perceivable difference, referred to as distance, between frames. Scenes with subtle detail or complex structural movements appear less convincing as video textures. Examples include river scenes, waterfalls and blowing grass. Better results are achieved from inputs such as a flying flag and flames where motion is fluid and in some way repetitive. The pendulum mentioned earlier also provides good results once the dynamics of movement are dealt with.
Movement between the different segments created in the analysis process nearly always results in some form of none contiguous attribute. The techniques indicated for use in the rendering stage help deal with this, but the results are not always suitable. It would seem that advancement in the quality of textures is likely to come from improved ability to cope with these inconsistencies and through the application of complex models to analyse training sequences (Schödl et al, 2000, pp 494 - 497; Phillips and Watson, 2003, pp. 10 - 14).

2.4. Advanced video textures

The methods used in rendering to help produce the required quality of video have further applications in developing more complex video textures. Videos of the same event from a number of angles can be warped together to create three-dimensional video textures. Pollard et al examine the how separate views can be process using edge detection then rendered into three-dimensional sequence (Pollard et al, 1998). Ideas such as background subtraction (Schödl et al, 2000, p497) and blue screen matting (Smith and Blinn, 1996) introduce the concept of video sprites. Here individual elements of a scene can be extracted and used to form video textures. The separate textures can then be recombined in a different context to provide a rich and dynamic scene.

2.5. Generalised media textures

Phillips and Watson have generalised the processes and methodologies associated with video textures to apply in the broader case of “media textures”. A logical step in relation to this is to look at how this generalisation applies to sound. We firstly encounter the fact that it is the “frequency composition of the audio wave which characterise it”. To derive meaning from the audio data it is separated into appropriate segments, and a Fourier Transformation is applied (Phillips and Watson, 2003, p. 10). This breaks down the audio wave into sine functions of time with different frequencies, totalling to the initial sound wave (Hoffman, 1997).

Analysis and synthesis is conducted in much the same manner as with video. On reaching the rendering stage some difficulties are caused by the continuous nature of audio waves, thus discrepancies tend to occur in the sound produced. These jumps between the recombined sound frames result in ‘clicks’ and a reduction in quality. In order to smooth these transition points, frames can be blended to create a new “composite frame”. This is used in place of the frame immediately
after the transition. Notably, such methodology is applicable to both audio and video and produces a noticeable improvement in output (Phillips and Watson 2003 p. 14).

2.6. Driving video textures

Possibly the most exciting application of video textures surrounds the introduction of external influences in controlling the texture. Schödl et al consider “user controlled terms” in what they call “video based animation”. Examples firstly address basic user interaction through the use of sliders. This is continued with implementation of mouse controlled textures along with a scripted path approach. In each case the element of user control acts as a constraint on frame selection. The distance measures derived in analysis are considered against a set of “user-specified frames” to determine appropriate transitions. Thus in the “synthesis stage” the video texture can be influenced through “parametric motion control” and reacts according the users input (Schödl et al, 2000, p496).

Driving video textures with speech relates closely to the work of Bregler, Covell and Slaney. “Video Rewrite” analyses existing audio through a system of phoneme labelling. The video element is examined by considering “visemes… the visual counterpart to phonemes”. This results in the ability to automatically create new footage of a person mouthing words not spoken in the original sequence (Bregler, Covell and Slaney, 1997). This work forms the closest precursor to that of video textures and certainly to this project as a whole (Schödl et al, 2000, p490; Phillips and Watson, 2003, p8).

2.7. Closing Discussion

Video and media textures have the potential for a wide range of application throughout many fields. The most apparent are areas such as visual effects, animation and the gaming industry. Consider the further possibilities surrounding user interaction driving textures and their horizon widens even further. In combination with systems such as “Immersive Video” (Moezzi et al, 1996) and advanced modelling techniques, video textures have the potential to produce astounding results.
Chapter 3
Analysis and Design

3.1 Application outline

The overall concept of a speech driven video texture talking head involves using video of a face with accompanying speech audio, referred to as training data (2.2.1.), to generate a new video texture (2.2). Separate input audio forms the foundation for this new video. The newly generated video, when combined with the input audio, should match coherently and should see the face in the output video speak or mouth along appropriately to the input audio.

This process can be broken down into two conceptual stages, training and execution. These are examined in more detail in sections 3.1.1 and 3.1.2 below.

3.1.1 Training

The training stage of the application involves analysis of training data and generation of rules for use in the execution stage. Figure 1 shows the main stages involved.

Figure 1

Training – An overview of the applications learning phase
Firstly the training data is split into audio and video for separate examination. The video is then broken down into individual frames and the audio divided into small segments each corresponding to a frame of video. From here individual frames of video and audio can be analysed.

The processing of each video frame involves deriving a representation of its content. Examination attempts to describe attributes of the facial expression or interpret key facial features. The procedure involves reducing the complexity of each frame and deriving data that characterises it. As with the frames of video, each frame length sample of audio undergoes analysis to determine its signature or major attributes. These will ultimately be compared with those of the input audio in the execution stage.

With simplified representations of the video and audio created the next stage of analysis is clustering (examined further in section 4.3.3.). The clusters created should reflect sets of frames with similar attributes, and thus similar underlying features. The centres of these clusters form the basis for the choice of frame in the execution phase. In this respect they represent rules for choosing video frames based on the attributes of audio.

### 3.1.2 Execution

The first task in the execution stage is the analysis of the input audio. This audio is treated in the same manner as the training audio. It is broken down into frames and analysed to produce a simplified data set representing each frames’ signature.

Figure 2

![Diagram](image-url)  
Execution – Training results and new audio are combined to generate a new video sequence
The next step is the generation of video from the analysed input audio. Firstly the cluster centres derived in training can be compared with the analysed audio input. For each audio frame the cluster exhibiting the best matching sound attributes is chosen. Based on the association between the video and audio from the training data, the cluster should contain video frames with features corresponding to the input audio. The choice of cluster creates a subset of video frames from which images can be chosen and used to create a small segment of output video. When all input audio has been dealt with we have the foundation to build the new video output.

Once a basic video is formed from the chosen frames, steps can be taken to improve continuity. Techniques such as blending and morphing (2.2.2.) will help reduce inconsistencies. Masking the transition between frames with such procedures should help produce a higher quality video and improve coherence with the audio. The final task in the execution stage is the combination of the new video sequence with the original input audio.

3.2 Approach and methodology

The nature of the prototype application being developed means many design decisions in the development process are depend on experimentation as well as trial and error. The applications’ inputs and outputs are well defined but a number of approaches are possible when analysing input and generating the video output. With this in mind implementation of the prototype is best approached in an iterative manner.

Simple methods and techniques used in early iterations can be built on and replaced with more complex approaches at each stage. Small sections of functionality can also be developed in isolation. This enables outcomes and quality of data at each stage to be determined. When elements of functionality perform as they should, they can be combined. This will result in a prototype application that in its first instance produces crude low quality results. From here on the application can be refined and refactored. At each step introduction of improved methods and more sophisticated techniques should improve output, ultimately resulting in a higher quality and more coherent final video.
3.3 Tools and applications

Two main options are available for the implementation of the prototype application. Coding could be done in C++. Pursuing this method makes use of my knowledge of the language and would enable the use of OpenGL of which I have experience. Using C++ also makes available the Leeds Real-Time Image Libraries. These provide a straightforward API for display and manipulation of images and video. Use of the libRTImage and libRTImageDisplay libraries would form a valuable code base and the foundation for much of the functionality needed in the prototype.

The other option is to use the application Matlab. Matlab has extensive support for the types of procedure required in the prototype. The image processing toolbox provides numerous functions for the reading, writing and manipulation of images. The statistics toolbox also implements many common algorithms and would provide straightforward access to clustering techniques. Furthermore Matlab provides built-in functions for the manipulation of sound files as well as the generation of Audio Video Interleaved (AVI) files.
4.1. Initial experimentation

The very first and fundamental element of the project is the capture of training data. Video and audio acquired from a Philips USB webcam feeds the training stage of the application. Thus decisions on setup and interface with this hardware form the first stage of implementation.

In the early stages of the project implementation was geared towards the use of C++. As a result initial work surrounded the capture and display of video using the Leeds Real-Time Image Libraries. Accomplishing operations such as the display of mpeg or QuickTime video and the capture of individual video frames from the webcam developed into small applications to allow the display of live video. From here consideration could be given to the refinement and processing of visual data.

Examination of captured images helped devise rudimentary ways of simplifying them and representing the facial image shown. The most apparent and straightforward way of achieving this was through thresholding. The thresholding of an image involves simplifying the pixel data held within it using a basic rule. This rule indicates a cut off point for a certain property or properties of the image. This property might be colour, brightness, saturation, or a combination of such attributes.

The output derived from the thresholding can easily be converted into a binary image. This representation considerably depletes the content of image but offers numerous options for further processing (Bracewell, 2004, p76 & p316).

Experimentation on a selection of images produced promising results. Figure 3 shows the raw image captured and figure 4 the result when a threshold is applied. We see that the threshold gives a simple representation of the darkest parts of the face (the areas show in white). These areas are also the major facial features. We see the eyes, nostrils and mouth, all indicated clearly in the processed images. Figures 5 and 6 show an open mouthed image rather than the closed mouth shown in figures 3 and 4. The large white area produced by an open mouth is clearly visible and provides a means to differentiate the content of the two images.
Although the thresholding process derived sensible and logical results a number of drawbacks became apparent. The quality of the output for a given threshold value varied substantially with different images. Images captured in different conditions required a threshold geared to their particular attributes. Choosing and maintain a particular threshold across a range of images gave unreliable results and often didn’t produce a useful outcome. It became clear that another methodology could prove more reliable in analysing the image data.

4.2. Transition to Matlab

At this early point in the implementation process it became clear that use of C++ would not be practical and that the methods and operations required in the application would be difficult to accomplish using this language. Use of the application Matlab had been an alternative considered in the initial analysis stage (3.3.). Exploring this option further made it clear that Matlab would provide a better environment in which to construct the application.
Matlab gives access a wide spectrum of built in functionality and support for a number of the processes that would have to be coded by hand in C++. The methods for image and sound processing attempted in Matlab proved more appropriate and achieved better results. These were retained for use in the final implementation and are detailed in the remainder of this chapter.

4.3. Analysis and training

The training stage described in section 3.1.1 uses image and sound processing techniques to gain meaningful representations of the training data. A training video of approximately two minutes is captured for analysis using the webcam. This length helps ensure sufficient data and variation within it, whilst avoiding excessive amounts of data slowing the training process and the algorithms used. Having obtained a length of training video the process starts with the extraction of the sound element, this is stored as a separate wave file (wav). The video is broken down into a series of jpeg image files each corresponding to a frame of video. This initial phase creates data readable by Matlab and in a suitable format for further analysis.

4.3.1. Image processing and representation

Image processing takes place on a frame by frame basis. To simplify the images and focus analysis an area of interest approach is used. The most appropriate region of the face relating to speech is clearly the mouth and this is used as the basis for decomposing the image. With input video specifically geared to ensure the face remains reasonable static, an assumption on the position of the mouth can be made. It should be the case that the mouth remains within the given region throughout the length of the training video. Thus for each 640 x 480 pixel image an area 160 x 80 pixels containing the mouth is isolated.

Image processing – The mouth region is isolated to focus analysis
To further compact the facial representation the intensity image is block processed. This involves breaking down the image into regions of 10 x 10 pixels. The mean of the pixel intensities within each block is calculated producing a single value for the region. The resultant interpretation of the mouth area is a 16 x 8 matrix of intensity values. Figure 8 depicts three sample frames from a training video alongside a visualisation are their processed representation.

Figure 8

Image representation – Original frames alongside their final intensity based decomposition
The final section of image processing transforms each intensity image into a vector. The vectors for each frame can then be combined into a matrix representing the entire visual element of the training sequence. Rows correspond to individual frames and each column a dimension of the intensity data.

Drawing from video texture literature, this representation provides the basis for evaluation of “frame-to-frame distances”. The removal of the mouth region alone also attempts to take advantage of improved performance achieved by “splitting the original video into regions and computing... for each region separately” (Schödl et al, 2000, p491). This coincides beneficially with use of the mouth as the coercive attribute behind the visual aspect of the texture.

4.3.2. Audio processing and representation

Sound processing obtains an audio signature associated with each frame of the training video. The first stage of this involves breaking the audio into sets of samples corresponding to individual frames of video. With predetermined attributes of 15 frames per second and an audio sample rate of 8000 Hertz, each frame of video coincides with approximately 533 audio samples. Thus the audio file is loaded and decomposed into sections (audio frames). Figure 9 shows a plot of an example audio frame taken from training data.

Figure 9

Audio frames – The audio is broken into short segments associated with each video frame
When attempting to derive meaning from the sound data we consider that methods for processing speech can be roughly generalised into two base approaches dealing with either “time-domain” or “frequency-domain” (Owens, 1993, p11). To examine the characteristics of each frame the later forms the basis of this analysis (see 2.5). Fourier transforms offer a “widely used technique for evaluating the frequency spectrum of speech” (Owens, 1993, p44). Through an “eminently physical relationship” theory shows that “waveforms and spectra are Fourier transforms of each other” (Bracewell, 1978, p1).

The digital signal captured above requires the use of a discrete Fourier transform (Magee, 2004, p44). This is applied using a fast Fourier transform algorithm. Each frame is thus separately processed to obtain its frequency spectrum. The result of the transform upon the audio frame shown in figure 9 is displayed below in figure 10.

**Figure 10**

As was the case with the image data it is desirable to reduce the complexity of the audio representation. To accomplish this, the audio frames are examined using a histogram count. This method indicates the number of observations for a given frequency range (the boundaries used are visible on the x axis of figure 11 below). The result gives a measure of distribution for the frequency
data obtained from the Fourier transform. The data created for the example audio frame (used in previous figures) is displayed below in figure 11

**Figure 11**

Audio representation – The audio frames are ultimately represented by the distribution of their frequency spectrum.

Each audio frame is now represented as 17 values indicating the distribution of its frequency spectrum. In much the same manner as with the visual data a matrix is constructed storing each frame and its dimensions. This forms the final representation of all audio training data.

### 4.3.3. Image and audio clustering

The final stage of the training process is the clustering of the image and audio data to develop collections of frames with similar attributes. The matrices constructed during image and audio processing are combined to produce a matrix containing a complete interpretation of the training data. Each row in this matrix therefore contains a series of values relating to both the visual and audio attributes of a frame from the training sequence.
This interpretation of the video and audio data provides a means to define a “measure of similarity” required in the clustering process (Anderberg 1973, p13). This similarity provides a foundation for determining a distance between frames analogous to that derived in the analysis stage of video texturing (Schödl et al, 2000, p491). Application of a clustering algorithm to the dataset results in the derivation of collections of frames with similar underlying properties. These properties are assessed by considering the proximity of each data point determined from the distance measure (Anderberg, 1973, p13). Euclidean or $L_2$ distance is used here for clustering, specifically to correspond with the measure used in “extracting the video texture” (Schödl et al, 2000, p491).

In this implementation clustering is achieved with the use of a K-means algorithm, used mainly due to its simplicity and accessibility. Non-hierarchical methods of clustering, such as K-means, are “designed to cluster data units… into a single classification of $k$ clusters” (Anderberg, 1973, p157). Anderberg indicates that “a substantial practical problem in performing a cluster analysis is deciding on the number of clusters in the data”. Thus examination is required to establish a suitable number of clusters ($k$). Matteucci further notes that “there is no general theoretical solution to find the optimal number of clusters for any given data set. A simple approach is to compare the results of multiple runs with different $k$ classes and choose the best one according to a given criterion” (Matteucci). In order to determine a suitable number of clusters into which to divide the frames we consider a measure of cost associated with the clustering process and evaluate it in the manner indicated by Matteucci.

**Figure 12**

Choice of $k$ – The number of clusters plotted against associated cost
Each frame is allocated to a cluster according to its position in a multidimensional space determined by its attributes (both audio and visual). It is desirable to determine a number of clusters whereby each frame exists an acceptable distance from its derived cluster centre. As a measure of cost we therefore consider the sum of all distances from data points to their associated cluster centre. This is assessed in turn for each number of clusters throughout a suitable range. A plot of this cost evaluation is shown in figure 12.

Initially as the number of clusters is increased the cost measure falls rapidly. The reduction in summed distances falls at each increment until a relatively consistent cost becomes apparent. Lower cost is desirable in that it indicates each frame is comparatively closer to a cluster centre. However, as a larger numbers of clusters are employed potential exists for the creation of empty clusters as well as the likelihood of repeated attributes within separate clusters. This trade off suggests an appropriate value for \( k \) should be chosen from the beginning of the low cost region evident in figure 12.

A value of forty clusters was initially chosen to characterise the training data. However, further investigation into the content of the clusters appeared to suggest that an unacceptable number of clusters displayed alike features. A reduced value of thirty clusters was ultimately chosen for implementation. Figures 13 and 14 depict the content of two representative clusters from the thirty. It is visible from these that the frames appear appropriately grouped and clusters contain suitably dissimilar representations.

Having evaluated a suitable number of clusters, the K-means algorithm is applied to the training data matrix. The output obtained indicates to which cluster each frame belongs and the central point of each of the thirty clusters. This data referred to in the analysis and design section (3.1.1.) as the cluster rules is the end product of the training stage and provides a means for frame choice when generating the output video.
Sample cluster – Frames are grouped based on both visual content and their associated audio.

Note the distinct pout evident in most frames. Not all frames exhibit this characteristic due to the influence of audio content.
Sample cluster – Each cluster should demonstrate distinct characteristics.
The content of the above cluster and that displayed in Figure 13 are noticeably dissimilar.
4.4. Input audio analysis

With the training phase complete focus turns to analysing the input audio for which video will be constructed. This is processed using the same approached as the training audio (4.3.2.). The audio is broken down into frames, its frequency spectrum is obtained and the distribution of this spectrum is determined. The result is data that is directly comparable to the training audio representation.

4.5. Video generation

Formulating the output video involves choosing a suitable video frame for each of the frames in the input audio. This is achieved through examination of the data produced during the training phase. The results of training are formulated to allow a hierarchical choice. Firstly the cluster best matching the audio frame is chosen. Then from within this cluster an appropriate frame can be selected.

4.5.1. Cluster choice

The clustering process essentially presents a set of implicit links to drive the video texture. This is a departure from the distance matrix approach encountered in other training routines which determine explicit links or transitions between frames (Schödl et al, 2000, p491; Phillips and Watson, 2003, pp 9-10). The clusters represent a constraint on our distance measure (4.3.3.) and impart a means to evaluate this distance and determine links dynamically. Presence within a certain cluster indicates both the suitability of a frame in relation to a particular audio signature and its correspondence with other frames within the cluster.

The links communicated in the clustered data are evaluated by considering the cluster centres produced by the K-means algorithm. Each cluster centre consists of dimensions for both video and audio. To conduct the comparison the dimensions associate with the audio representation are extracted. The attributes of each input audio frame can then be accessed against this proportion of the cluster centre. The cluster centre nearest to the audio frame provides the best match and implies a subset of frames for use in the video sequence.
4.5.2. Frame choice

In production of a system that meets minimum requirements, no specific means need be used to select a frame from within the chosen cluster. An arbitrary frame chosen from each cluster provides a basic starting point to produce video output. This frame should prove sufficiently representative of the cluster and is used whenever its associated cluster is chosen.

4.6. Video construction

The previous procedures in the execution phase yields a series of references to frames from the training video. The remaining routine retrieves the actual images from the training sequence and combines them into a finished video. The initial version of the application produces a video that combines the frames without any additional processing. This video, built to meet minimum requirements, draws from the thirty frames chosen to denote each of the thirty clusters and places them in the order determined by input audio analysis.

Construction of the output is completed by combining the original input audio with the generated video. The resultant AVI can then be played using a standard media player.

4.7. Enhancements

A number of avenues were indicated to enhance the application (1.5). The extensions pursued focus on improving the quality of the video sequence produced.

4.7.1. Improved frame choice

The first enhancement implemented deals with obtaining a frame from the chosen cluster (4.5.1) by a more appropriate means than the initial arbitrary choice. A single frame per cluster is still used however the frame closest to the centre point of the cluster is determined. This ensures the frames used in the video sequence are truly representative of the content of their cluster. Figure 15 displays a sample set of these cluster centre frames.
Figure 15

Central Frames – An example set of 30 frames selected to represent their cluster based on frame to cluster centre distances
4.7.2. Video cross-fading

This enhancement addresses techniques for “disguising discontinuities in the video texture” (Schödl et al., 2000, p494). Movements between images in the video sequence often create abrupt changes and jumps. Video cross-fading aims to reduce these noticeable transitions by blending neighbouring images. This creates a new “composite frame” which replaces an existing frame in the sequence (Phillips and Watson, 2003, p14).

There are numerous possibilities for combining groups of images to smooth the video output. However, the main concern with any approach is to retain the timing of the original sequence and avoid distortion (Phillips and Watson, 2003, p14). To accomplish this we ensure the major proportion of any cross-faded image comes from the current image it represents in the video.

The first method of cross-fading implemented combines the current image (i) with a proportion its’ subsequent image (i +1), replacing the current image in the sequence. This is summarised in the equation below, here \( \alpha \) represents the desired contribution of subsequent image ranging from 0 to 1.

\[
i = (1 - \alpha)i + \alpha i_{+1}
\]

4.7.3. Two way cross-fading

Extending the cross-fading procedure further, the next implementation undertaken uses a two way cross-fading method. In this approach the smoothing is applied using both the subsequent (i +1) and preceding (i -1) frames in the series. Again the central image (i) must be retained the as the major contributor. i is then replaced with the new composite image.

\[
i = \alpha i_{-1} + (1 - 2\alpha)i + \alpha i_{+1}
\]
Chapter 5
Evaluation

5.1 Prototype appraisal

Assessment of the prototype involved conducting a demonstration during which output sequences were produced for a variety of input audio. Both the training and input audio attempted to avoid inappropriate background noise and capture was conducted in similar conditions. The input audio drew largely from a predetermined vocabulary on which the application was trained, though a number of sequences were produced using audio outside this set. The vocabulary used aimed to include a variety of short and long phrases and a range of vocalisations.

Evaluation of the output is not a straightforward matter and in general approaches issues of a qualitative nature. It is also worth noting the lack of any “accepted metrics for evaluating lip-synced footage” (Bregler, Covell and Slaney, 1997, p358). A number of data sets were produced during the demonstration in an attempt to evaluate the output through a more quantitative means. However, the appraisal mostly relies on observation of the output produced. Thus in examining the output we consider two main faculties: the synchronisation between audio and video and the nature of articulation evident in facial and mouth motions.

In assessing synchronisation we wish to determine how well the output video responds and if transitions between video frames coincide with changes in audio. Whether speech produces a timely reaction is an essential element of the application and equally the absence of speech should produce an appropriate result. Examination of articulation considers the quality of motion produced in the video. We wish to assess whether mouth movements are appropriate and life-like. At a basic level, does the mouth open during speech? Does it then close after the period of vocalisation? If this is achieved we can further investigate whether individual words and sounds are dealt with suitably. Realism and a natural appearance could be considered the main criteria for assessment of articulation.
5.1.1 Audio Synchronisation

In order to meet minimum requirements (1.3), the main area of functionality required in the prototype, is the production of a response to the input audio. This essential evaluates to whether appropriate frames are introduced into the video sequence during speech in the audio input.

Comparing graphical representations of the input audio with the video generated provides us with the ability to directly assess the correspondence of the video sequence. In doing this we require a measure of response or change to characterise the frames of the video sequence. Figures 16b and 17b below depict changes in the average intensity of the pixels within the frames of the output video produced. Each is displayed alongside a plot of the audio sequence from which it was generated (16a and 17a).

Figure 16

Synchronisation – Comparing the waveform of an input audio sequence a) with pixel intensity changes in the corresponding video output b) indicates the high quality of synchronisation achieved in the output sequence.
Figure 17

Noise – The background noise highlighted in a) results produces an undesirable reaction in the video sequence b)

The changes in intensity see in figures 16b and 17b indicate movement to different images throughout the video sequence. Figures 16 and 17 display an excellent correspondence between the audio track and image transitions and this is typical of the output video produced. The application reacts appropriately introducing different video frames in response to the audio signal. Quiet sections of audio are also matched with periods of constant intensity indicating that a particular image or still frame (used during periods without speech) is correctly maintained throughout. Performance in audio synchronisation proved to be equally good for phrases outside the training set.

The main issue encountered, with regard to synchronisation, arose from false movements in reaction to background noise and none speech sound. The regions highlighted in figure 17a show such areas of noise and in figure 17b the reaction produced in the video sequence. These sounds generally produce the display of an inappropriate frame for a brief period. Although in some cases a suitable still frame
is displayed. Never the less, noise affects the video sequence. This is undesirable, but could be elevated with greater control over input capture conditions.

5.1.2. Articulation

By our nature “we are very sensitive to the synchronization between speech and lip motions” (Bregler, Covell and Slaney, 1997, p353). Thus achieving convincing articulation even for a limited vocabulary is not a trivial task. Articulation is heavily dependent on synchronisation. If the video created doesn’t respond suitably to the audio then even a basic level of articulation is unlikely to be achievable or indeed convincing. Considering the vast range of facial movements used in speech as well as the subtle nature of many of these motions it was not deemed suitable to specify requirements (minimum or otherwise) surrounding articulation.

From examination of the output sequences it becomes evident that the prototype performs well at a basic level of articulation. The mouth opens and closes with a reasonable degree of consistency with the audio track, this is visible in figure 18 below. The distinct spikes in intensity plot are caused by the introduction of frames containing a darker open mouth region (displayed in frames 9 and 49). The articulation achieved by the prototype can certainly not be considered realistic, but mouth movements are timely, and correspond well to individual words and sounds. At some points it is also evident that the mouth moves appropriately for particular sounds, forming shapes that correspond favourably with the vocalisation. However, this occurs only intermittently and is contrast by other instances where unsuitable shapes and mouth movements occur.

Much of the smooth and flowing motion of the original training video is lost due to the use of a limited number of frames. Mouth movements in the output video are abrupt and tend to lack fluidity. However, the video frames used at key points in relation to the audio track are suitably correspondent and the overall motion of words are retained. This is displayed especially well when training audio is reused as input. Doing so allows direct assessment of the applications ability to choose frames that correspond with those in the training set. Figure 18 displays this process using the training phrase ‘How are you?’. Using three key points in the phrase, the ‘ho’ in how (frame 9), the ‘oo’ in you (frame 49), and the sequences still frame (frame 24) we can observe the consistency between the original and generated video. The three example frames from the original training video are displayed with their corresponding output frames. The frames from the output video are visibly a good match for the original training frames. This ability to choose images that are satisfactorily consistent with those in the original sequence generally ensures the new video has a pleasing overall appearance.
It became evident that articulation is superior for short words and sounds. The video produced for more complex words with multiple syllables is notably less convincing. This is also the case with maintained vocalisations such as long vowel sounds. However, it should be noted that such sounds were not well addressed within the training vocabulary. The most noticeable problem with articulation arises from the chosen still frame, displayed during periods without speech. Ideally the still frame should involve a relaxed closed mouth. The frame used, although featuring a closed mouth, was often not wholly desirable and reduced realism of the sequence. This situation is visible in frame 24 of figure 18 above.
5.1.3 Frame choice and articulation

The improved method of frame choice implement as an extension (4.7.1) produces superior results for articulation. Although the general quality of correspondence between speech and mouth movements remains relatively unchanged, two facets are noticeably improved. Obtaining frames from the centre of the cluster achieves a marked reduction in the occurrence of unsuitable mouth shapes. This approach also produces an improved choice of still frame. Although not completely satisfactory, the frame used is typically more convincing than that of the original methodology.

5.1.4 Video dynamics

Moving away from assessment of the content of video and the speech driven aspect of the application, we now consider the quality of the video texture in its own right and how the cross fading techniques employed contribute to this. Examining changes in intensity again provides a basis to assess continuity between images.

![Graph](image)

**Figure 19**

Target quality - Plotting average pixel intensity within the frames of the training sequence provides a useful comparison when evaluating the output sequences
J. B. Moxon

It is firstly worth examining output from the original training footage to determine how the continuity measure reacts. Results for the first two hundred frames of the demonstration training sequence are displayed in figure 19.

Figure 20

Random image choice – Worst case results for intensity analysis would likely resemble the random image choice shown above, therefore this too provides another important benchmark for evaluation

In contrast to figure 19, the worst possible outcome of the video texturing process could be considered comparable to random image choice. Intensity values for a sequence of two hundred randomly chosen images are show in figure 20. As could be expected, intensities in the training sequence are considerably less erratic. Adjacent frames are generally less than 0.5 of a unit apart and movements over a greater intensity range typically take place during a series of frames. This results in flowing transitions rather than the large jumps evident in figure 20. Clearly the main facet the generated video should avoid are these sharp changes across more than one of intensity unit.
Figure 21 shows the intensity values for three typical video sequences produced as part of the demonstration. The intensity changes visible in these examples are often over a range greater than one unit and are visibly more pronounced than those in figure 19. However, encouragingly the movements are considerably less erratic than those in figure 20. Observing the video sequences, the majority of transitions between frames do cause noticeable inconsistencies. Jitters and jumps in the content of the frames are often evident and this reduces the realism of the textures.

The sequences shown in figure 21 were created using the initial video construction methodology where frames are combined without additional processing. Cross-fading aimed to improve continuity between frames and reduces the undesirable aspects noted above. Figure 22 shows the intensity representation of the same three sequences after applying two way cross-fading.
It is visible from Figure 21 that cross-fading is moderately successfully in helping to smooth intensity changes between frames. Transitions can be seen to be more curved and shifts in intensity are less pronounced. It is also evident from examination of the above representations, as well as playback of the video sequences, that there is no perceivable detriment to synchronisation. Cross-fading does however introduce the problem of blurring. The jumps and jitters of the unprocessed sequence are manifested as blurriness due to the combination of dissimilar frames. This in itself is not particularly problematic, however, the transitions between sharp and blurry are easily noticeable and are not conducive to realism.
5.2 Conclusion

In examination of the prototype and the output produced consideration has been given to major facets and required features of the video texture. Regarding synchronisation between audio and video, it is evident that changes in audio along with the presence and absence of speech are identified and produce a coordinated and corresponding response in the video created. Extending this to consider articulation we have seen that these responses are also appropriate. In this respect the images used in reaction to the audio are suitably fitting and provide a sensible overall effect.

The quality of the video itself suffers from a bias towards the speech driven aspect of the application. Although the cross-fading techniques employed address this to a certain degree the results are reasonable but not convincing. A trade-off between control and continuity is a notable issue (Schödl et al, 2000, p496) and something that could have been addressed better within the design and implementation of the application. It also became apparent that the original training sequence exerts a substantial influence on the quality of the final video. Many of the more extreme discrepancies and notable inconsistencies in the videos produced are a result of difficulties obtaining a satisfactory training sequence. Camera movements and changes in lighting conditions proved detrimental to the output of the application, although this was partly due to assumptions and implementation decisions made surrounding image analysis.

Other work, notably “Video Rewrite” (Bregler, Covell and Slaney, 1997) and the concepts reviewed by Bailly et al, presents numerous alternative and advanced methods applicable to the subject area approached by this project. Through the enhancements suggested in section 1.4 it had been hoped possible to address these more sophisticated techniques. However, in hindsight and within a limited timeframe these were unfortunately beyond the scope of this project. In summary, review of the application suggests that the techniques, tools used as well as overall methodology of work proved suitable and obtain an agreeable overall outcome.

5.3 Extensions

By the nature of this project and the application produced, a varied array of opportunities exist for further work. Production and evaluation of the prototype highlighted a number of issues that further work could address. These along with avenues evident in relevant literature are considered below.
The applications reliance on training video quality is certainly an area for possible improvement and enhancement. Techniques such as brightness equalisation and “video stabilization” could provide a means to improve the sequence prior to analysis (Schödl et al, 2000, p491). Equally, further work might wish to address different methods of image analysis and representation, alleviating this issue altogether. The region dependant approach chosen in image analysis doesn’t account for movement or alteration in the rest of the scene. Thus a logical progression would be to consider more of the facial area and possibly the scene as a whole. This would certainly provide more data and the potential to produce a more convincing texture.

Improved use of the training data is another element enhancements might address. The current output suffers as a result of the use of a limited number of frames from the training set. Further work might approach the application of within cluster frame transitions along with improved movement between clusters. In a wider context further work might examine adhering more faithfully to the methodology of video textures with a view to improving video quality and dynamics. On the other hand extensions might focus on the audio analysis aspect of the application and produce a more detailed means of audio representation. The relationship between phonemes and visemes (potentially a project in its own right) appears an interesting avenue in this respect.

It is immediately obvious that video textures present a diverse spectrum of opportunities for further work. Driving video via user interaction is certainly an aspect that future investigation may wish to address further. The generalisation of the texturing process detailed by Philips and Watson also introduces addition options and potential for extensions to approach the use of other types of media. The work presented in this report addresses what is clearly a rich and varied topic combining aspects from a number of disciplines of computing, thus enhancements could certainly be tailored towards a variety of fields of research.
J. B. Moxon

References


D. Magee (2004), AI21 – Speech, Audio and Image Processing (Course Notes), School of Computing, University of Leeds


J. B. Moxon


Appendices

Appendix A – Reflection

This project was a considerable undertaking and required a lot of commitment and perseverance. I am pleased with the overall outcome of the project and I hope my time and effort reflects in this report and the deliverables I have produced.

The production of the report and work involved in the project was quite different from the usual style of assessment I had encountered in the Computer Science element of my degree. Fortunately the Economics proportion of my Joint Honours degree made me better prepared and comfortable with the style of work. I can understand why some (especially single honours) students could be ill prepared for such an undertaking, especially the report element.

In the early stages of work I felt somewhat out of my depth. It was at times not easy to see where the project was going and I found it difficult to apply myself. This was largely down to the initial choice of implementation tools. The choice had been made with my previous experience and existing skills in mind. As a result the approach didn’t best suit the nature of the work to be undertaken. Fortunately I began experimentation and basic implementation at a sufficiently early stage to be able to switch tools and use an application more suited to my needs. After an initial difficult period learning how to use the new tools, I began to feel more comfortable with the project and I could work with increased confidence.

Frequent meetings with my project supervisor played a large part in overcoming difficulties and provided an opportunity to clarify issues and improve my understanding. I feel I developed a good working relationship with my supervisor and this was a major influence in the success of the project. I would certainly stress the importance of communication and contact time with supervisors to students due to undertake a final year project. Weekly meetings provided the opportunity to set intermediate goals and ensure work is conducted steadily and methodically.

Time management is a crucial part of all University work and particularly important with a large piece of work like a final year project. It is certainly the case that my work could have been better organised and my planning wasn’t as thorough as it could have been. However, I do feel that I worked in a measured and adequately planned manner. I can vouch for the importance of not putting off work, and
working at least loosely to a schedule. Doing this ensured the problem I encountered with changing implementation methods didn’t cause the difficulties it could have.

In hindsight meeting intermediate deadlines, such as the submission of the mid-project report and draft chapters, was extremely helpful. Actively reviewing my work and producing items suitable for submission, rather than rough work and notes, helped aid my understanding and brought to mind issues and difficulties. The work produced also proved invaluable in my final write up, and helped spread the work load.

Overall I can’t say my final year project was a totally enjoyable experience, although a number of aspects of it were rewarding and it is satisfying to see something you’ve worked hard on come together. I do, however, understand the value of completing a project. Having undergone the process I can see its significance in demonstrating an ability to learn and organise such a considerable assignment.
Appendix B – Operation Instructions

Setup

1. Copy the entire content of the demo directory (provided on CD) to a suitable location.

This directory contains training audio and images along with the Matlab scripts that make up the prototype.

2. Run Matlab ensuring the chosen location, containing the data and scripts, is set as the working directory.

Training

3. Load the provided training data by calling the read function at the Matlab prompt:
   ```matlab
   >> [images, audio] = read;
   ```

4. Invoke the training sequence as follows:
   ```matlab
   >> [audio_centres, central_frames] = train(images, audio);
   ```

The 3 Matlab variables: audio_centres, central_frames and images form the complete training results set. They may be exported as a Matlab file and retained for reuse in the remaining steps.

Execution

For the creation the talking head video, audio input is required in mono ‘wav’ file format with a sample rate of 8000 Hz at 16 Bits.

5. Generate a talking head Matlab movie from ‘wav’ file using the following command:
   ```matlab
   >> movie = generate(audio_centres, central_frames, images, '<wavfile>');
   ```

Where <wavfile> is the full location of the audio file to be used. e.g. './audio.wav' (including speech marks).

6. Finally the Matlab movie created above can be exported to an ‘avi’ file using the built in movie2avi function:
   ```matlab
   >> movie2avi(movie, '<filename>');
   ```
Where <filename> is the full location of the desired ‘avi’ file e.g. './output.avi' (include speech marks)

Steps 5 and 6 may be repeated as required with different audio files.