[Ste96] *Ralf Steinmetz;* **Human Perception of Jitter and Media Synchronization;** erscheint in IEEE Journal on Selected Areas in Communications, Februar 1996.

Human Perception of Jitter and Media Synchronization

Ralf Steinmetz

IBM European Networking Center

Vangerowstraße 18 • 69115 Heidelberg • Germany

Phone: +49-6221-59-3000 • Fax: +49-6221-59-3300

steinmetz@vnet.ibm.com

Abstract: Multimedia synchronization comprises both the definition and the establishment of temporal relationships among media types. The presentation of 'in sync' data streams is essential to achieve a natural impression, data that is 'out of sync' is perceived as being somewhat artificial, strange, or even annoying. Therefore the goal of any multimedia system is to enable an application to present data without no or little synchronization errors. The achievement of this goal requires a detailed knowledge of the synchronization requirements at the user interface. This paper presents the results of a series of experiments about human media perception that may be used as 'Quality of Service' guidelines. The results show that skews between related data streams may still give the effect that the data is 'in sync' and gives some constraints under which jitter may be tolerated. We use our findings to develop a scheme for the processing of non-trivial synchronization skew between more than two data streams.

1 Introduction

In accordance with [Stei93][StNa95] we understand multimedia in the following way: A multimedia system is characterized by the integrated computer-controlled generation, manipulation, presentation, storage, and communication of independent discrete and continuous media. The digital representation of data and the synchronization between these various data are the key issues for integration. Synchronization is needed to ensure a temporal ordering of events in a multimedia system.

For single data streams a stream consists of consecutive logical data units (LDUs). In the case of an audio stream, LDUs are individual samples or blocks of samples transferred together from a source to one or more sinks. Similarly with video, one LDU may typically correspond to a single video frame and consecutive LDUs a series of frames. These have to be presented at the sink with the same temporal relationship as they were captured giving so called "intra-stream" synchronization.

The temporal ordering must also applied to related data streams, where one of the more common relationships is the simultaneous playback of audio and video with 'lip synchronization'. Both media must be 'in sync' otherwise the result will not be adjudged as satisfactory. In general, "inter-stream" synchronization involves relationships between all kinds of media including pointers, graphics/images, animation, text, audio, and video. In the following discussion, 'synchronization' always refers to "inter-stream" synchronization. As human perception varies from individual to individual it is usual in subjunctive experiments to carry out experiments with a sample of individuals to obtain a reasonable cross-section of results.

To reach the goal of an error-free data delivery, audio, video and other data are often multiplexed (i.e. physically combined in one data unit) at the source and demultiplexed at the sink. Multiplexing is not always wanted or possible, as different media need to be handled by different adapters in a system, however handling of previously related data leads to time lags between the media streams. These lags have to be adjusted for at the sink in order to produce an 'in sync' presentation. Some work on multimedia synchronization mechanisms was done in [AnHo91] [BLHM92] [LKGe92] [LLKG93] [LKGe94][ShSa90] [Stei92], as well as topics devoted to define synchronization requirements [LiGh90a] [LiGh90b] [Nico90] [Ravi92] [Stei90].

It is often reported that audio can be played up to 120 ms ahead of video and conversely video displayed up to 240 ms ahead of the audio. Both temporal skews are noticed, but can be accepted by the user without any significant loss of effect. Some authors however report a tolerance of only +/-16 ms [Dann93] to be acceptable.

The lack of in-depth analysis of synchronization between the various kinds of media and, in particular lip and pointer synchronization led us to conduct some experiments of our own to obtain results that allow us to quantify the quality of service requirements for multimedia synchronization.

The remainder of this text is organized into ten sections, Section 2 outlines the structure of the lip synchronization experiments, the respective results are given in Section 3 and 4. Section 5 presents the results on pointer synchronization and remaining types of media synchronization are discussed in Section 6. The aggregation of various individual media synchronization results are analyzed in Section 7, Section 8 defines and summarizes the results in terms of the required quality of service parameters. In Section 9 the results of human perception of jitter are presented and finally the appendix includes an example of the questionnaire used by test participants as well as all results in graphic form.

2 The Lip Synchronization Experiment

'Lip synchronization' refers to the temporal relationship between an audio and video stream for the particular case of humans speaking. The time difference between related audio and video LDUs is known as the 'skew'. Streams which are perfectly 'in sync' have no skew, i.e., 0 ms. We conducted experiments and measured the skews that were perceived as 'out of sync'. In our experiments users often mentioned that something is wrong with the synchronization, but this did not disturb their feeling for the quality of the presentation. Therefore, we additionally evaluated the tolerance of the users by asking if the data out of sink affects the quality of the presentation (see also the questionnaire in Appendix B).

In discussions with experts that work with audio and video we came to realize that generally subjects responded to or remembered particular parts of the clips, therefore a wide range of skews (up to 240 ms) we observed. A comparison and a general usage of these values are somewhat doubtful because the environments from which they resulted were not comparable. In some cases we encountered the 'head view' displayed in front of some single color background on a high resolution professional monitor whereas in others a 'body view' in a video window at a resolution of 240*256 pixels in the middle of some dancing people. In order to get accurate and good skew tolerance levels we selected a speaker in a TV news environment in a head and shoulder shot (Figure 1). In this orientation the viewer is not disturbed by background information and the viewer should be attracted by the gesture, eyes, and lip movement of the speaker.

Our study was performed in the news environment in which we recorded the presentation and then re-played it with artificially introduced skews created with professional editing equipment skewed at intervals of 40ms i.e. -120ms, -80ms, -40ms, 0ms, +40ms, +80ms, +120ms. Steps of 40 ms were chosen for:

- (1) the difficulty in human perception to distinguish any lip synchronization skew with a higher resolution.
- (2) the capability of multimedia software and hardware devices to refresh motion video data every 33ms/40ms

Figure 1: Left: Head View, Middle: Shoulder View, Right: Body View

Each test lasted approximately 45 minutes which constituted a session, with 3 sessions for the 3 views, the same text was used for each view. The order of the sessions had no effect, which was verified. Individual sequences with different skews were shown in random order. The results are shown within Table 2 in the Appendix.

We deduced that 30 seconds of video is sufficient for getting the users impression. Some candidates, that were more experienced with video technology and synchronization issues, 5 seconds was sufficient to identify the introduced skew, if any.

The sample size chosen was 107 and they were selected as fairly as possible across all ages and both sexes. To have a representative sample we did not take into account habits, like the time spent watching TV, or the social status or any other characteristics of the test candidates.

Unfortunately we were only able to perform the test with a few candidates not knowing the purpose of the exercise. The scenario being, once a test candidate noticed a synchronization fault they would be shown no further clips, as they then noticed the mismatch immediately, yielding results for unexpecting subjects. The result being what people perceive as 'out of sync' is more tolerated when the candidate does expect there to be synchronization skew. Therefore more of the unexpecting subjects classified the clip as 'in sync'.

In order to provide results for building multimedia systems for all types of users we have to assume that a user will make frequent use of such a system and interact for a longer time with the application. Therefore, the results of users being aware of possible synchronization faults provide the correct basis.

3 Results: Detection of Lip Synchronization

Figure 2 provides an overview of the results. The vertical axis denotes the relative number of test candidates who detected a synchronization error, regardless of being able to determine if the audio was before or after the video. Our initial assumption was that the three curves related to the different views would be very different, but as shown in Figure 2 this is not the case.

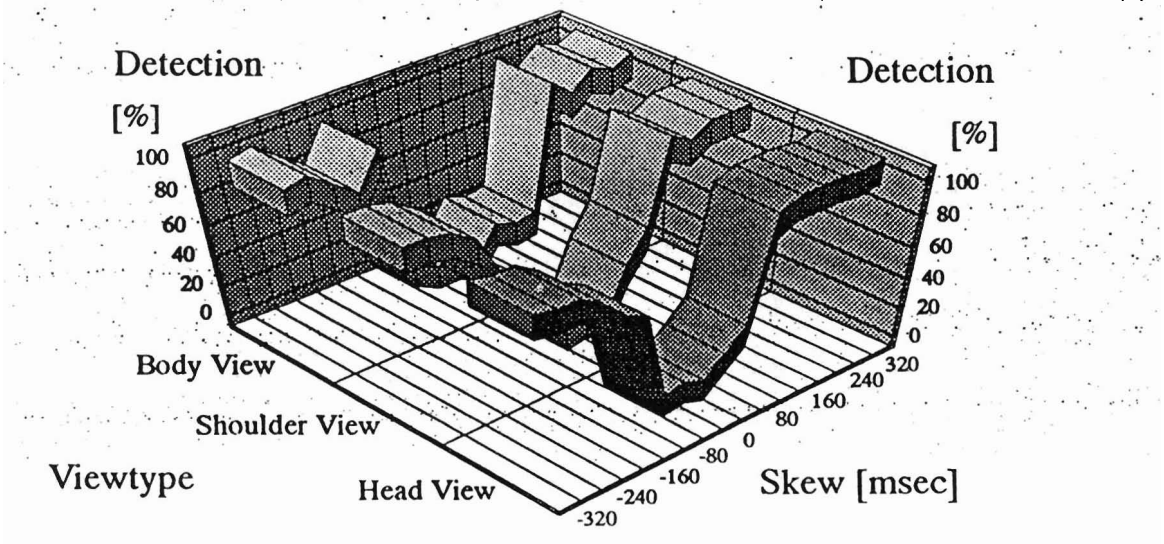

Figure 2: Detection of synchronization errors with respect to the three different views. Left part, negative skew; video ahead of audio; right part, positive skew; video behind audio

Figure 3 shows the same curves in more detail. A careful analysis provides us with information regarding the asymmetry, some periodic ripples and minor differences between the various views.

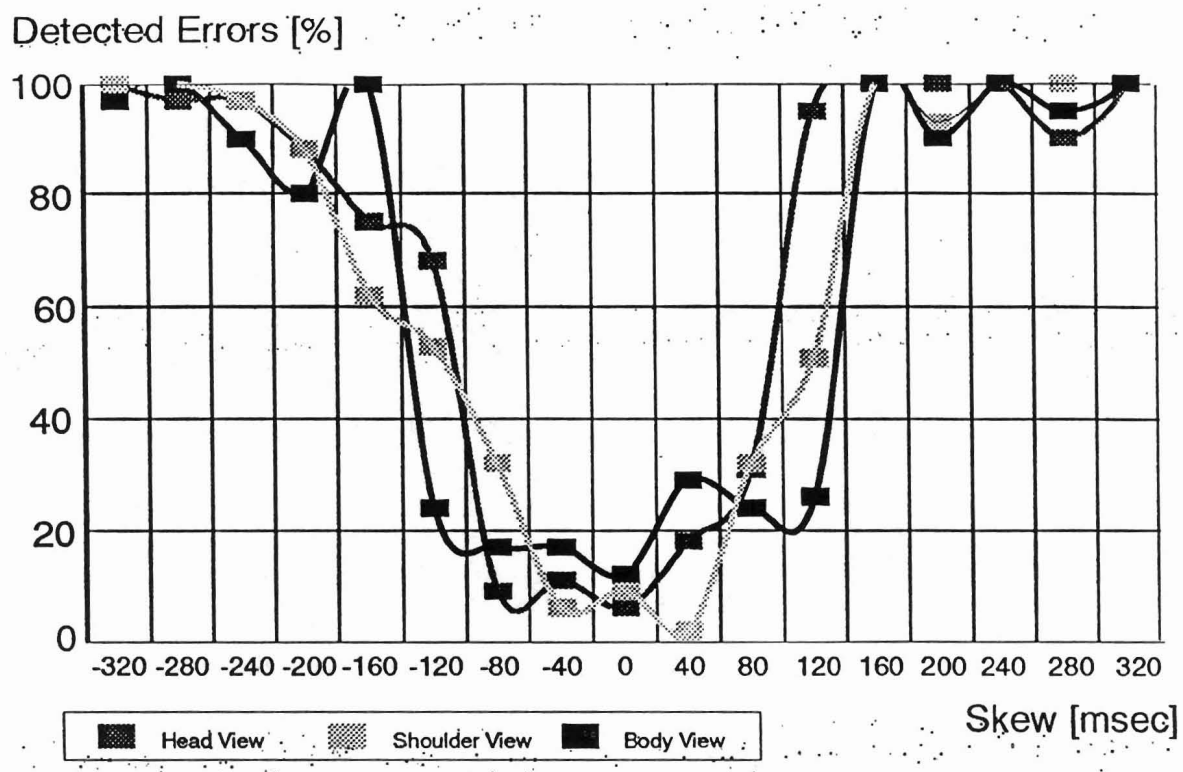

Figure 3: Detection of synchronization errors

Left of the central axis the graph relates to negative skew values where the video is ahead of the audio and on the right where the audio is ahead of the video. Day to day we often experience the situation where the motion of the lips are perceived a little before the audio is heard, due the greater velocity of light than sound, this is indicated by the right hand side of the curves being steeper than the left sides.

The 'body view' curve is broader than the 'head view' curve as at the former a small skew is easier to notice. The 'head view' is also more asymmetric than the 'body view', due to the fact that the further away we are situated, the less noticeable an error is.

At a fairly high skew the curves show some periodic ripples; this is more obvious in the case where audio is ahead of video. Some people obviously had difficulties in identifying the synchronization error even with fairly high skew values. A careful analysis of this phenomenon is difficult due to the sample volume (few more than a 100), the media content to be synchronized and the human mind and mood. However, one plausible explanation could be: At the relative minima, the speech signal was closely related to the movement of the lips which tends to be quasi periodic. Errors were easy to notice at the start, end of pauses as well as whenever a change in tone is introduced (a point being emphasized). Errors in the middle of sentences are more difficult to notice. Also we tend to concentrate more at the start of a conversation than once the subject is clear. A subsequent test containing video clips with skews according to these minima (without pauses and not showing the start, the end, and changes in tone) caused problems in identifying if there was indeed a synchronization error.

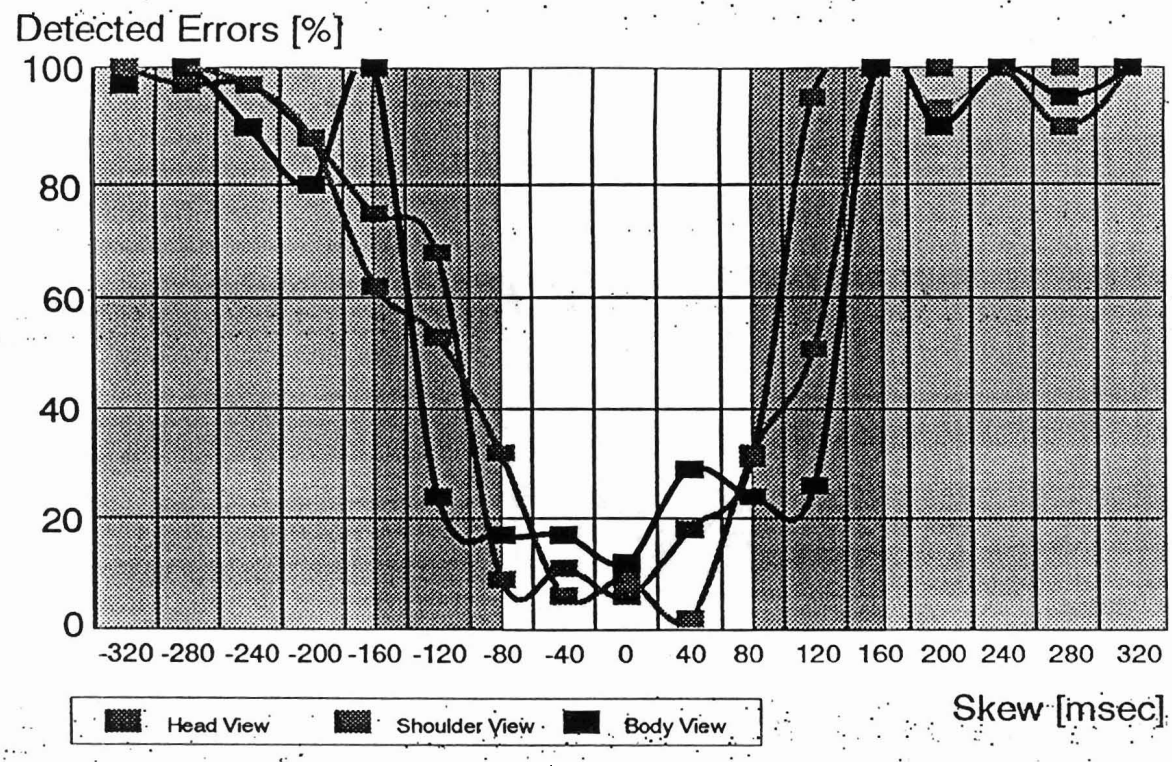

Figure 4: Areas related to the detection of synchronization errors

Figure 4 shows the following areas:

- The **'in sync' region** that spans a skew between -80 ms (audio behind video) and +80 ms (audio ahead of video). In this zone most of the test candidates did not detect the synchronization error. Very few people said that if there was an error it did affect their notion of the quality of the video. Additionally, some results indicated that the perfect 'in sync' clip was 'out of sync'. Our conclusion is that lip synchronization can be tolerated within these limits.

  The **'out of sync' areas** span beyond a skew of -160 ms and +160 ms. Nearly everybody detected these errors and were dissatisfied with the clips. Data delivered with such a skew is in general not acceptable. Additionally, often a distraction occurred; the viewer/listener became more attracted by this 'out of sync' effect than by the content itself.

- In the **'transient' area** where *audio is ahead of video*, the closer the speaker was the easier errors were detected and described it as disturbing. The same applies to the overall resolution, the better the resolution was the more obvious the lip synchronization errors became.

- A second **'transient' area** where *video is ahead of audio* is characterized by a similar behavior as above as long as the skew values are near the in sync area. One interesting effect did emerge and it was that video ahead of audio could be tolerated better than the vice versa. As above the closer the speaker is, the more obvious the skew is.

  This asymmetry is very plausible: In a conversation where two people are located 20 m apart, the visual impression will always be about 60 ms ahead of the acoustics due to the fast light propagation compared to the acoustic wave propagation. We are just more used to this situation than the ones in the test.

We obtained similar results when using the hammer with nails clip although the transient areas were narrower (also we used only 10 test candidates). In this experiment the view had little influence. The presentation of a violinist in a concert as well as a choir did not show more stringent skew demands than a speaker.

A comparison using different languages namely English showed no difference. Some minor experiments with Spanish, Italian, French and Swedish verified that the specific language has almost no influence on the results.

We did not find any variation between groups of participants with different habits regarding the amount of TV and films that they watched. Also we looked at the affect of the speed of audio, no difference was detected between the same person speaking in a fast, a normal or a slow manner.

Professionals cutters and TV related technical personnel showed a smaller level of skew tolerance, as expected. When they detected an error they could correctly state if audio is ahead of or behind video. A sitting with professional video cutting teams showed similar results. One out of three professionals stated that she/he would recognize an error with 40ms, all mentioned that they would recognize a 'lip sync error' of 80ms, but however this might not influence the quality of the perceived information.

4 Results: Quality of Lip Synchronization

Figure 3 and Figure 4 outlined the perception of synchronization errors. Just as important as the error itself is the effect of such an 'out of sync' video clip has on perception. Therefore the test candidates were asked to qualify a detected synchronization error in terms of being acceptable, indifferent, or annoying (see Question 3 at Appendix B). Out of these answers we derived the 'level of annoyance' graph, Figure 5.

The envelope curve (the upper edge of the dark area) defines the amount of candidates who detected a synchronization problem This is the same curve for the 'shoulder view' as shown in Figure 4 and Figure 5 without a spline interpolation.

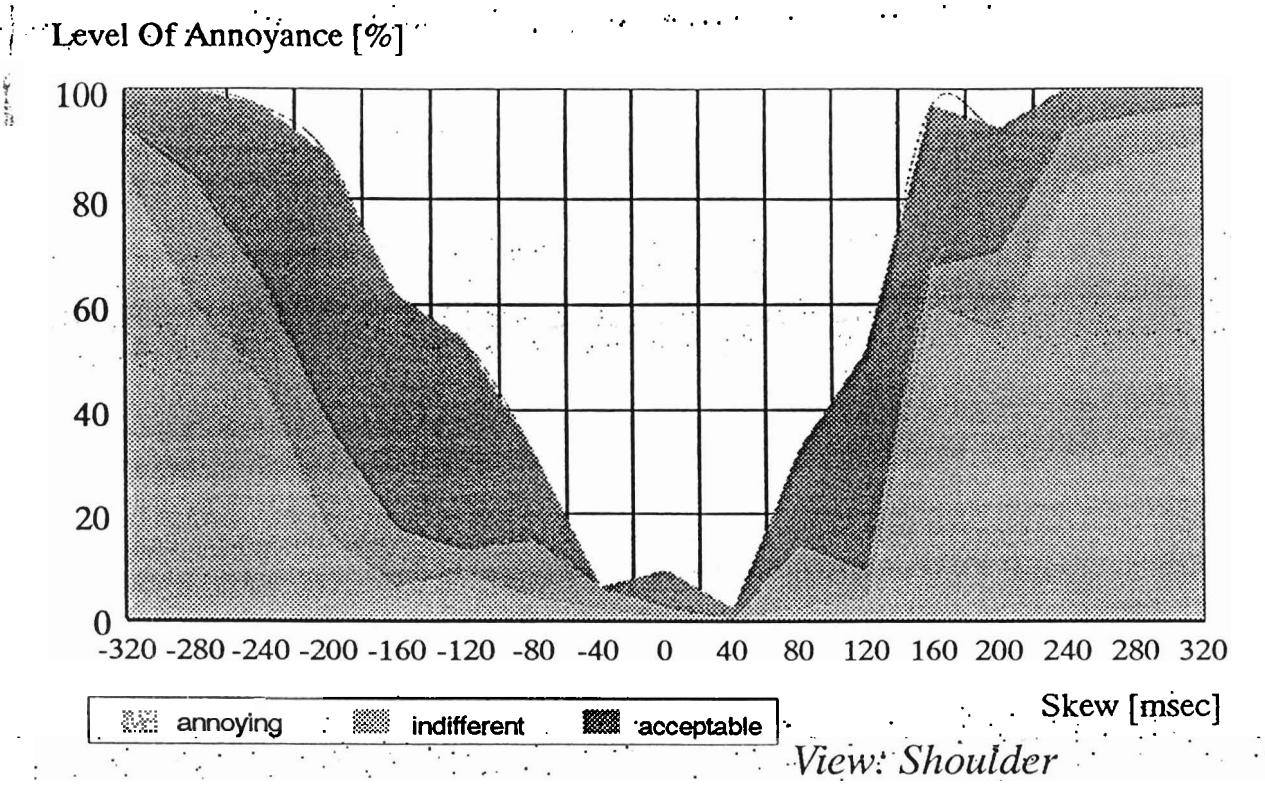

Figure 5: Level of annoyance at shoulder view

The dark grey areas relate to all test candidates who detected a synchronization error and found clip watchable with this synchronization error. In a small follow-on experiment we selected a few test candidates who would tolerate such a skew and showed them a whole movie with a -160 ms skew where the video was ahead of the audio. There were little annoyances reported soon after the start of the film all candidates concentrated on the content instead of being attracted by the synchronization offset. The curve at the bottom of the dark grey area shows an asymmetry between sound and light as mentioned before.

The light grey area indicates the people who found the skew distracting. During the evaluation phase of this study on synchronization we introduced a skew of +80 ms and -80 ms into two whole movies which were shown to a few candidates who found it irritating but still could concentrate on the content. The same experiment however with a skew of -240 ms or +160 ms would lead to a real distraction from the content and to a severe a feeling of annoyance.

We can therefore conclude that skews of between -80 ms and +80 ms are deemed acceptable by most casual observers.

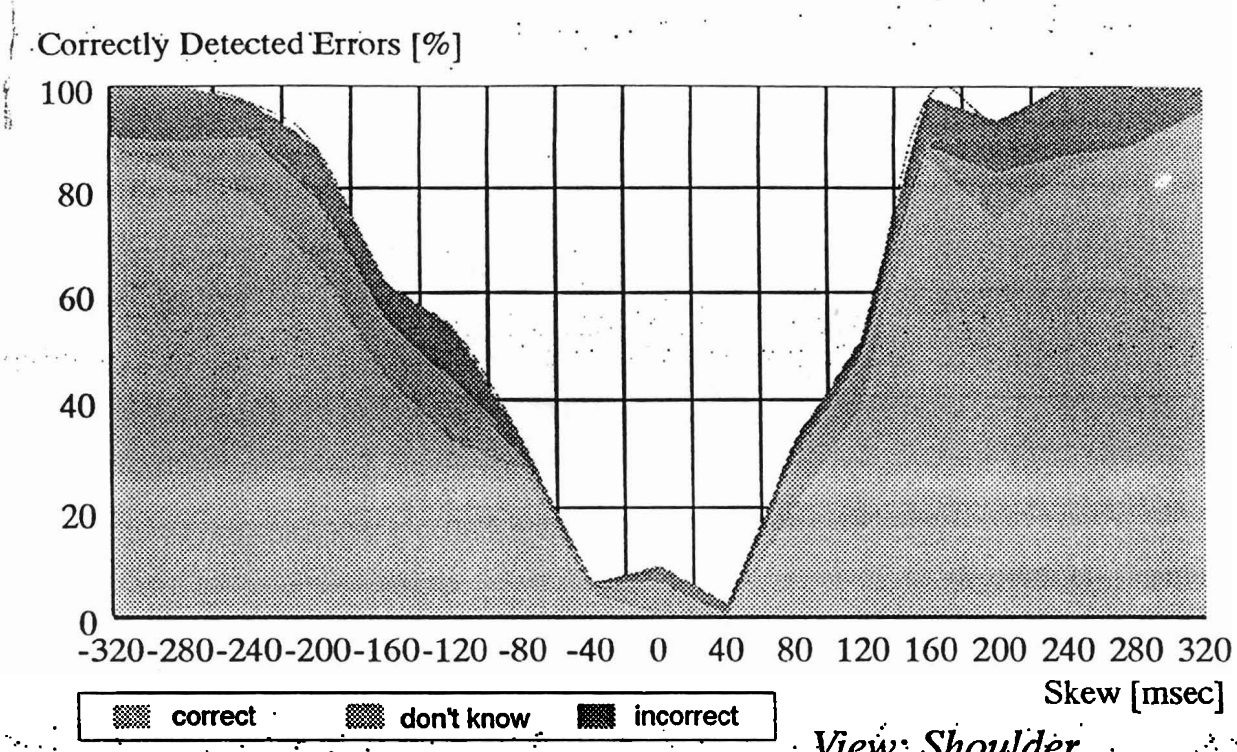

Figure 6: Correct detection of the perceived synchronization error

In order to double check the candidates were asked to define exactly which type of synchronization error they noticed. As stated before it is easier just to detect that something is wrong rather than to state if audio is ahead of video or vice versa. Figure 6 summarizes the results of correct perception of the skew in the 'shoulder view' scenario.

Figure 6 shows the number of people who detected a synchronization problem, this is in fact the same curve for the 'shoulder view' as shown in Figure 3 and Figure 4 without a spline interpolation. The lowest envelope curve of the light grey area represents the number of people who detected a mismatch of audio and video.

As can be expected near the error-free synchronization value (0 ms) it was difficult to determine the type of skew, however large skew values were often identified correctly.

5 The Pointer Synchronization Experiment and Results

In a computer-supported co-operative work (CSCW) environment, cameras and microphones are usually attached to the users' workstations. In our next experiment we looked at a business report that contains some data with accompanying graphics. All participants have a window with these graphics on their desktop where a shared pointer that is used in the discussion. Using this pointer speakers point out individual elements of the graphics which may be relevant to the discussion taking place. This obviously requires synchronization of the audio and the remote telepointer.

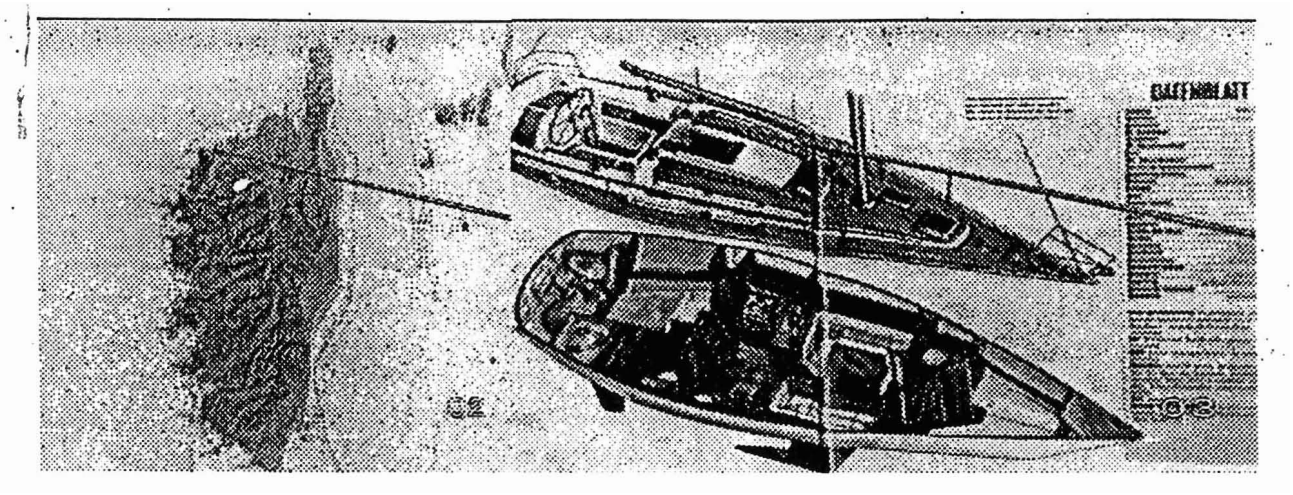

Figure 7: Pointer synchronization experiment based on a map and on a technical sketch

We conducted two experiments:

- The first was to explain some technical parts of a sailing boat while a pointer locates the area under discussion (Figure 7, right side). The shorter the explanation, the more crucial the synchronization turns out to be therefore we selected a fast speaking person who was to use fairly short words.
- Additionally we held a second experiment with the explanation of a travelling route on a map (Figure 7, left side) this involves the continuous movement of the pointer.

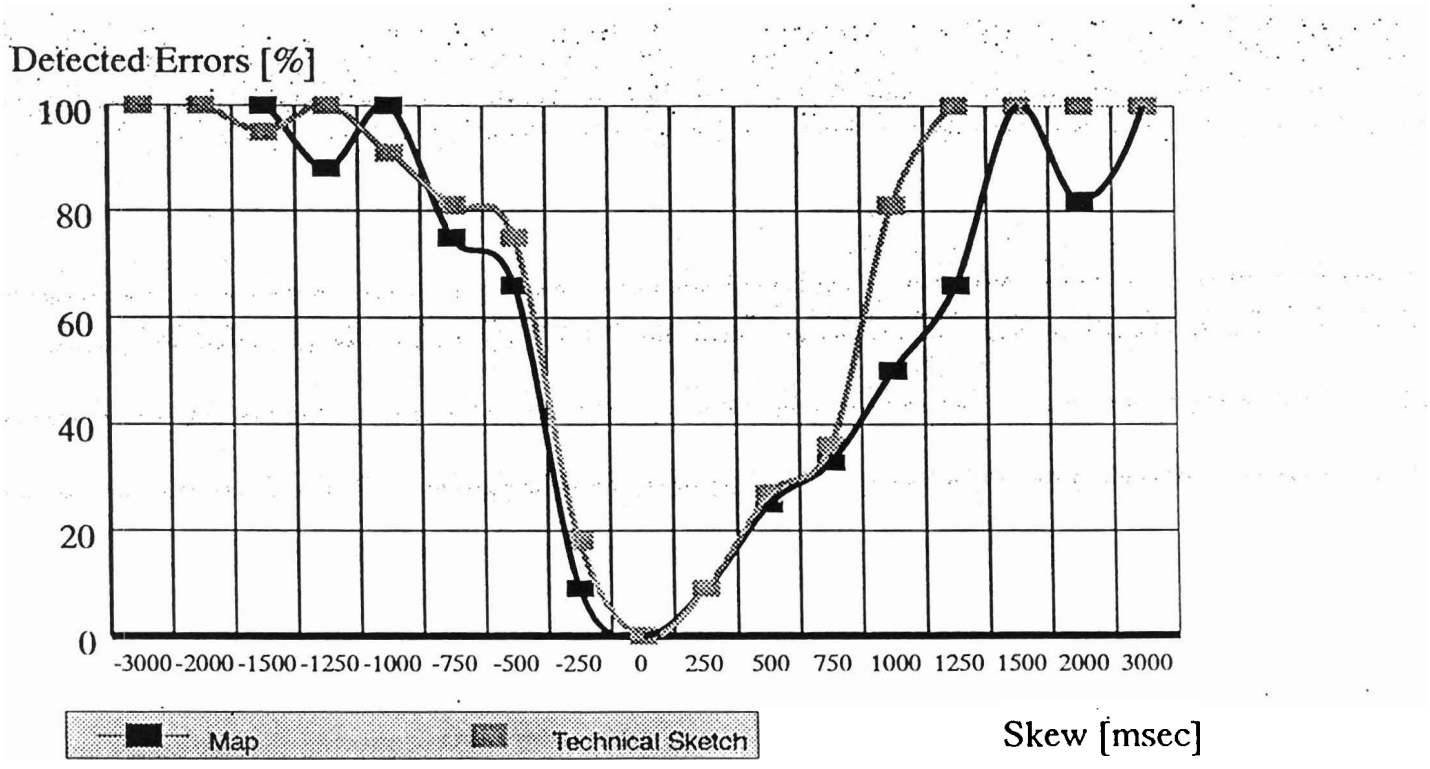

Figure 8: Detection of the pointer synchronization errors

From the human perception point of view, pointer synchronization is very different from lip synchronization as it is much more difficult to detect the 'out of sync' error at skew values near the error-free case. While a lip synchronization error is a matter of discussion for a skews between 40 ms and 160 ms, for a pointer the values lie between 250 ms and 1500ms; figure 8 shows the some results.

Using the same judgement technique as in our first experiments, the **'in sync' area** related to audio ahead of pointing is 750ms and for pointing ahead of audio it is 500 ms. This zone allows for a clear definition of the 'in sync' behavior regardless of the content.

The **'out of sync' area** spans a skew beyond -1000 ms and beyond +1250 ms. At this point the test candidates began to mention that the skew makes the attempted synchronization worthless and became distracted unless the speaker slowed down or moved the pointer more slowly. From the user interface perspective, this is not acceptable quite clearly the practise of pointing to one location on the technical figure while discussing another is virtually impossible.

In the **'transient' area** we found that many test candidates noticed the 'out of sync' effect but it was not mentioned as annoying. This is certainly different from 'lip sync' where the user is more sensitive to the skew and without question found it annoying.

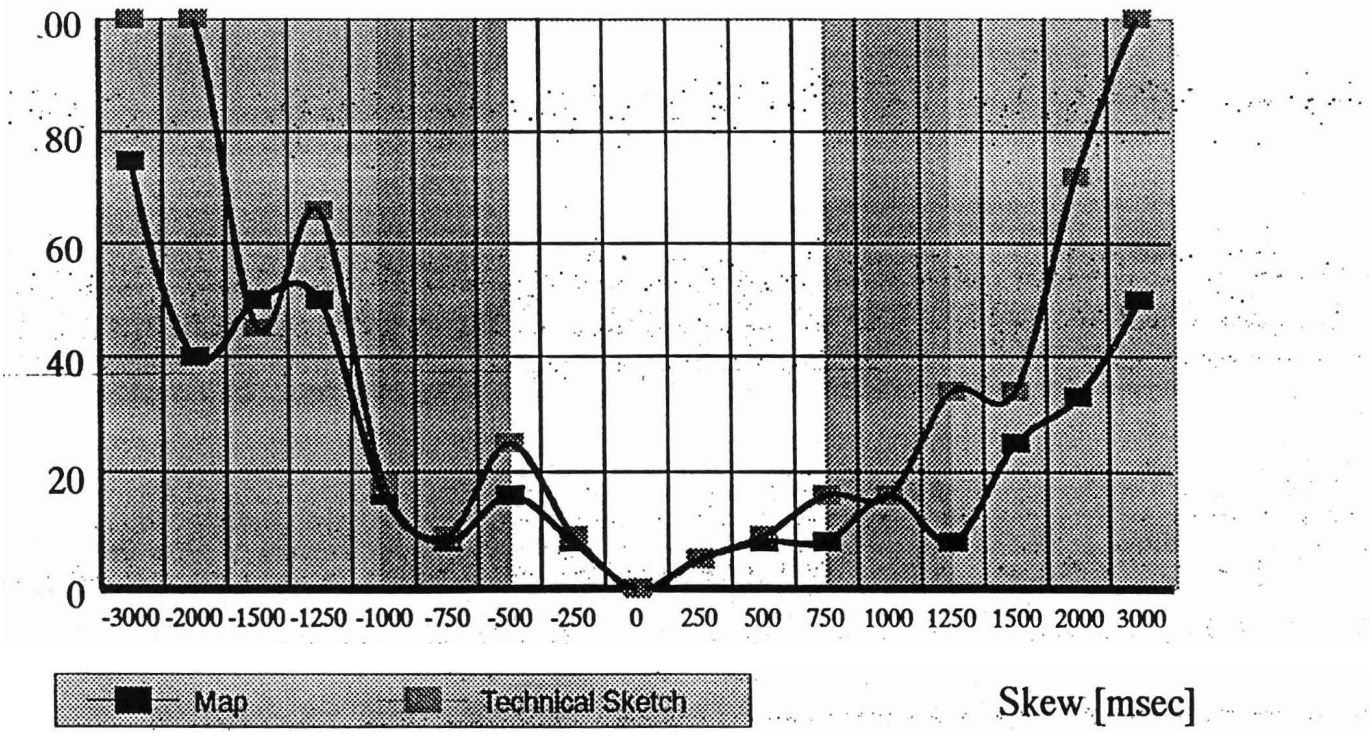

Figure 9: Level of Annoyance of the pointer synchronization errors

Figure 9 shows the number of people who disliked or are indifferent towards the pointer synchronization error. It is worth mentioning that for several skew values most of the test candidates detected the fault but did not object to such a skew, hence the broad "in sync' and 'transient' area.

6 Elementary Media Synchronization

Lip synchronization and pointer synchronization were investigated due to inconsistent results from available sources. The following summarizes other synchronization results to give a complete picture of synchronization requirements.

Since the beginnings of digital audio the 'jitter' to be tolerated by dedicated hardware has been studied. Dannenberg provided us some references and the following explanations of these studies: In [Bles78] the maximum allowable jitter for 16 bit quality audio in a sample period is 200ps, which is the error equivalence to the magnitude of the LSB (least significant bit) of a full-level maximum-frequency 20 KHz signal. In [Stoc72] some perception experiments recommended an allowable jitter in an audio sample period between 5 and 10 ns. Further perception experiments were carried out by [Lick51] and [Wood51], the maximum spacing of short clicks to obtain fusion into one continuous tone was given at 2ms (as cited by [RuAv80]).

The combination of *audio and animation* is usually not as stringent as lip synchronization. A multimedia course on dancing, for example, could show the dancing steps as animated sequences with accompanying music. By making use of the interactive capabilities, individual sequences can be viewed over and over again. In this particular example the synchronization between music and animation is particularly important, experience showed that a skew of +/- 80ms fulfills the user demands despite some possible jitter. Nevertheless, the most challenging issue is the correlation between a noisy event and its visual representation, where a case could be the simulated crash of 2 cars. Here we encounter the same constraints as for lip synchronization, +/- 80 ms.

Two audio tracks can be tightly or loosely coupled, the effect of related audio streams depends heavily on the content:

- A stereo signal usually contains information about the location of the sources of audio and is *tightly coupled.* The correct processing of this information by the human brain can only be accomplished if the phases of the acoustic signals are delivered correctly. This demands for a skew less than the distance between consecutive samples leading to the order of magnitude of 20 µs. [DaSt93] reports that the perceptible phase shift between two audio channels is 17µs. This is based on a headphone listening experiment. Since a varying delay in one channel causes the apparent location of a sound's source to move, Dannenberg proposed to allow an audio sample skew between stereo channels within the boundaries of +/- 11µs. This is derived from the observation that a one-sample offset at a sample rate of 44kHz can be heard.
- *Loosely coupled* audio channels are a speaker and, e.g., some background music. In such scenarios we experience an affordable skew of 500 ms. The most stringent loosely coupled configuration has been the playback of a dialogue where the audio data of the participants originate from different sources. The experienced acceptable skew was 120 ms.

The combination of *audio with images* has its initial application in slide shows. By intuition a skew of about 1s arises which can be explained as follows [Dann93]: Consider that it takes a second or so to advance a slide projector. People sometimes comment on the time it takes to change transparencies on an overhead projector, but rarely worry about automatic slide projectors.

A more elaborated analysis leads to the time constraints equivalent to those of pointer synchronization. The affordable skew decreases as soon as we encounter *music* played in correlation *with notes* for, e.g., tutoring purposes. [Dann93] points out that here an accuracy of 5 ms is required: Current practice in music synthesizers allows delays ranging up to 5 ms, but jitter is

The synchronized presentation of *audio with some text* is usually known as audio annotation in documents or, e.g., part of an acoustic encyclopedia. In some cases the audio provides further acoustic information to the displayed or highlighted text in terms of 'audio annotation'. In an existing 'music dictionary', an antique instrument is described and simultaneously played. An example for a stronger correlation is the playback of a historical speech of, e.g., J.F. Kennedy with simultaneous translation into a German text. This text is displayed in a separate window and must relate closely to the actual acoustic signals. The same applies to the teaching of a language where in a playback mode the spoken word is simultaneously highlighted. Karaoke systems are another good example of necessary audio and text synchronization.

For this type of media synchronization the affordable skew can be derived from the duration of the pronunciation of short words which last in the order of magnitude of 500 ms. Therefore the experimentally verified skew of 240 ms is affordable

The synchronization of *video and text* or *video and image* occurs in two distinct fashions:

- In the *overlay mode*, the text often is an additional description to the displayed moving image sequence. In a video of playing billiard, the image is used to denote the exact way of the ball after the last stroke. The simultaneous presentation of the video and the overlaid image is important for the correct human perception of this synchronized data. The same applies to a text which is displayed in conjunction with the related video images: Instead of having the subtitles always located at the bottom, it is possible to place text close to the respective topic of discussion. This would cause an additional editing effort at the production phase and may not be for the general use of all types of movies but, for tutoring purposes some short text near by the topic of discussion is very useful. In such overlay schemes, this text must be synchronized to the video in order to assure that it is placed at the correct position. The accurate skew value can be derived from the minimal required time. A single word should appear on the screen in order to be perceived by the viewer: 1 s is certainly such a limit. If the media producer wants to make use of the flash effect, then such a word should be on the screen for at least 500 ms. Therefore, regardless of the content of the video data we encounter 240 ms to be absolutely sufficient.
- In the second mode *no overlay* occurs, skew is less serious. Imagine some architectural drawings of medieval houses being displayed in correlation with a video of these building: While the video is showing today's appearance, the image presents the floor plan in a separate window. The human perception of even simple images requires at least 1 s, we can verify this value with an experiment with slides: the successive projector of non-correlated images requires about 1 s, as the interval between the display of a slide and the next one in order to catch some of the essential visual information of the slide. A synchronization with a skew of 500 ms (half of this mentioned 1 s value) between the video and the image or the video and text is sufficient for this type of application.

Consider the billiard ball example from before: a video shows the impact of 2 billiard balls and the image of the actual 'route' of one of the balls is shown by an animated sequence. Instead of a series of static images, the track of the second ball can be followed by an animation which displays the route of the ball across the table. In this example any 'out of sync' effect is immediately visible. In order for humans to be able to watch the ball with the perception of a moving picture, this ball must be visible in several consecutive adjacent video frames at a slightly different positions, an acceptable result can be achieved if every 3 subsequent frames the ball moves by it's diameter. A smaller frame rate may result in the problem of continuity as often seen in tennis matches on the television. As each frame last about 40ms and 3 subsequent frames are needed, an allowable skew of 120 ms would be acceptable. This is very tight syn-

chronization figure which has been suitable for the examples we looked at. Other examples where video and animation are being ever more combined is that of computer generated figures in films.

Multimedia systems also incorporate the real-time processing of *control data.* Telesurgery is a good example where graphical information is displayed based on readings taken by probes or such like instruments. No overall timing demand can be stated as these issues highly depend on the application itself.

7 Aggregation of Media Synchronization

So far, media synchronization has been evaluated as the relationship between two kinds of media or separate data streams. This is the canonical foundation of all types of media synchronization. In practice, we often encounter more than two related media streams; a sophisticated multimedia application scenario incorporates the simultaneous handling of various sessions. As an example is a video conference where a window displays the actual speaker and the audio emerges from an attached pair of speakers, the application is the explanation of new space command station.

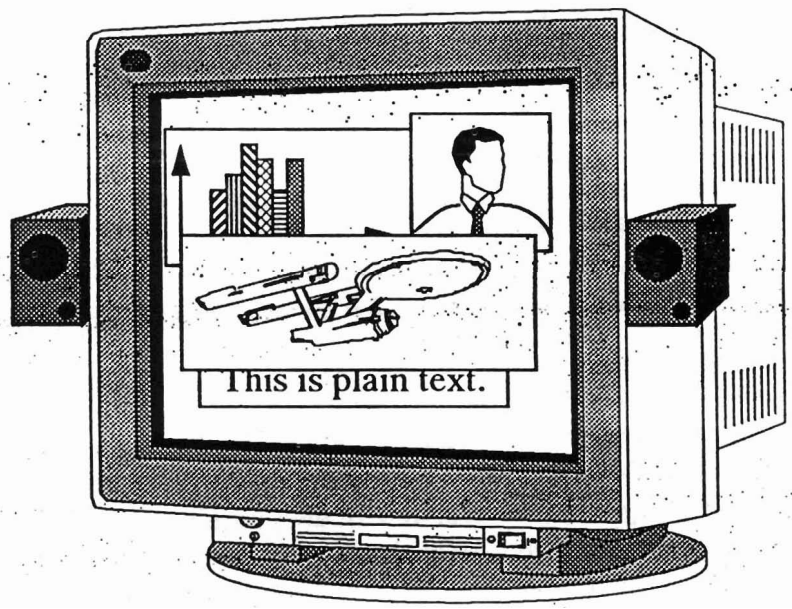

Figure 10: Aggregation of media at the user interface

Video and audio data are related by the lip synchronization demands. Audio and the telepointer are related by the pointer synchronization demands. The relationship of video data and the telepointer is then yielded by a simple combination. In this example we will define the following skews:

max skew (video ahead_of audio) = 80 ms
max skew (audio ahead_of video) = 80 ms
max skew (audio ahead_of pointer) = 740 ms
max skew (pointer ahead_of audio) = 500 ms

leading to the skew

skew (video ahead_of pointer) =< 820 ms
skew (pointer ahead_of video) =< 580 ms

In general these requirements can be derived easily by the accumulation of the canonical skew as shown in the above example. The information gathered by the aggregation of media is of interest for the user as well as for the multimedia system which must provide service according to these values.

In some cases exist too many specifications of a synchronization skew; for example a language lesson that includes audio data in English and Spanish as well as the related video sequence. The course builder enforces lip synchronization between video and audio regardless of the language (+-80ms). Additionally the sentences need to be synchronized in order to switch from one language to the other, we chose a figure of 400ms for this case. As lip synchronization is more demanding than the synchronization between the languages, this would lead to the following skew specification:

1. max skew (video ahead_of audio_english) = 80 ms
2. max skew (audio_english ahead_of video) = 80 ms
3. max skew (video ahead_of audio_spanish) = 80 ms
4. max skew (audio_spanish ahead_of video) = 80 ms
5. max skew (audio_english ahead_of audio_spanish) = 400 ms
6. max skew (audio_spanish ahead_of audio_english) = 400 ms

This specification consists of a set of related requirements in which all of them need to be fulfilled, i.e. we have to find 'the greatest common denominator'. For each canonical form, the derived skews are computed:

1+2+3+4:
max skew (audio_english ahead_of audio_spanish) = 160 ms
max skew (audio_spanish ahead_of audio_english) = 160 ms

1+2+5+6:
max skew (video ahead_of audio_spanish) = 480 ms
max skew (audio_spanish ahead_of video) = 480 ms

3+4+5+6:
max skew (video ahead_of audio_english) = 480 ms
max skew (audio_english ahead_of video) = 480 ms

In the second step the most stringent set of all requirements are selected:

1. max skew (video ahead_of audio_english) = 80 ms
2. max skew (audio_english ahead_of video) = 80 ms
3. max skew (video ahead_of audio_spanish) = 80 ms
4. max skew (audio_spanish ahead_of video) = 80 ms
5. max skew (audio_english ahead_of audio_spanish) = 160 ms
6. max skew (audio_spanish ahead_of audio_english) = 160 ms

The following step any set of synchronization requirements can be chosen from the above derived calculations:

max skew (video ahead_of audio_english) = 80 ms
max skew (audio_english ahead_of video) = 80 ms
max skew (audio_english ahead_of audio_spanish) = 160 ms
max skew (audio_spanish ahead_of audio_english) = 160 ms

In summary, the above procedures allow us to solve two related problems:

- If the applications impose a set of related synchronization requirements on a multimedia system, we are now able to find out the most stringent demands.
- If a set of individual synchronization requirements between various data streams is provided, we are now able to compute the required relationships between each individual pair

of streams.

Both issues arise in non-trivial systems when estimating, computing or negotiating the quality of service as it is outlined in the next section.

8 Synchronization Quality of Service

The control of synchronization in distributed multimedia systems requires a knowledge of the temporal relationship between media streams. Synchronization requirements can be expressed by a quality of service (QoS) specification, one QoS parameter can define the acceptable skew within the concerned data streams, namely, it defines the affordable synchronization boundaries. The notion of QoS is well established in communication systems, in the context of multimedia, it also applies to local systems. If audio and video parts of a film are stored as different entries in a database, lip synchronization according to the above mentioned results should be taken into account.

In this context we want to introduce the notion of *presentation* and *production level synchronization*:

- *Production level synchronization* refers to the QoS to be guaranteed prior to the presentation of the data at the user interface. It typically involves the recording of synchronized data for subsequent playback. The stored data should be captured and recorded with no skew at all, i.e. "in sync". This is particularly applicable if the file is stored in an interleaved format. At the participant's site the actual incoming audiovisual data is 'in sync' according to the defined lip synchronization boundaries. Assuming the data arrives with a skew of +80 ms and let audio and video LDUs be transmitted as a single multiplexed stream over the same transport connection then it will be displayed apparently "in-sync". Should the data be stored on the harddisk and presented simultaneously at a local workstation and to a remote spectator then for correct delivery the QoS should be specified as being between -160 ms and 0 ms. At the remote viewer's station without this additional knowledge of the actual skew the outcome might be that by applying these boundaries twice, data is not 'in sync'. In general, any synchronized data which will be further processed should be synchronized according to a production level quality, i.e. with no skew at all.
- The experiments discussed in this report identifies *presentation level synchronization*, it defines whatever is reasonable at the user interface. It does not take into account any further processing of the synchronized data; presentation level synchronization focuses on the human perception of synchronization. As shown in the above paragraph, by recording the actual skew as part of the control information, the required QoS for synchronization can be

easily computed.

Media		Mode, Application	QoS
video	animation	correlated	+/- 120 ms
	audio	lip synchronization	+/- 80 ms
	image	overlay	+/- 240 ms
		non overlay	+/-500 ms
	text	overlay	+/- 240 ms
		non overlay	+/-500 ms
audio	animation	event correlation (e.g. dancing)	+/- 80 ms
	audio	tightly coupled (stereo)	+/- 11 µs
		loosely coupled (dialog mode with various participants)	+/- 120 ms
		loosely coupled (e.g. background music)	+/- 500 ms
	image	tightly coupled (e.g. music with notes)	+/- 5 ms
		loosely coupled (e.g. slide show)	+/- 500 ms
	text	text annotation	+/- 240 ms
	pointer	audio relates to showed item	-500 ms, + 750 ms[1]

Table 1: Quality of Service for synchronization purposes

1. pointer ahead of audio for 500 ms, pointer behind audio for 750 ms

The required QoS for synchronization is expressed as the allowed skew. The QoS values shown in Table 1 relate to presentation level synchronization. Most of them result from exhaustive experiments and experiences, others are derived from literature as referenced. To our understanding, they serve as a general guideline for any QoS specification. During the lip and pointer synchronization experiments we learnt there are many factors which can influence these results. We understand this whole set of QoS parameters as first order result to serve as a general guidance, these values may be relaxed depending on the actual content.

9 Perception of Jitter

So far we have looked at synchronization as being "interstream synchronization", i.e., at the relationship between LDUs of two or more different data streams. However, synchronization is also important in the context of "intrastream-synchronization", i.e., denoting the relationship between LDUs within one data stream.

In any distributed system we experience a delay between a packet being sent and the same packet being received, this is known as the end-to-end delay. In asynchronous networks this delay varies. Jitter is defined to be the maximum difference between end-to-end delays experi-

enced by any two consecutive packets [ZhKe91]. Hence jitter implies a varying packet (and LDU) rate at the receiver. Jitter can either introduce gaps in the continuous playback of data streams or it shortens the playback of some LDU (a group of audio samples or a video frame).

Multimedia systems try to avoid any jitter in audio and video data streams wherever possible; mechanisms are conceived for an continuous presentation at the user interface. However, as the user does not perceive every variance to be disturbing and some may even go unnoticed, we looked at what a user really perceives as being an error-free data presentation while the presentation itself contains some kinds of temporal errors.

Jitter in packetized **audio** transmission is commonly addressed by buffering at the presentation site. The first packet is artificially delayed at the receiver for the period of the control time in order to buffer sufficient packets to provide for continuous playback in the case of presence of jitter.

In the case of playing audio, and in particular voice data, all experiments showed that glitches are immediately detected by any listener. Voice data is known to consist of talking and silent periods, naturally jitter in silence intervals are not perceived as error by the listener. Since talkspurts are generally isolated from each other by relatively long silence periods, voice protocols typically impose the control time on the first packet of each talkspurt. In this case, the 'slack time' of a packet is defined as the time difference between its arrival time at the receiver and its playback time [DLWe93]. This is the point in time at which playback of the packet must begin at the receiver in order to achieve a zero-gap playback schedule for the talkspurt. Due to jitter, a packet may arrive before or after its playback time. In the former case, the packet is placed in a queue, the packet voice receiver queue, until it is due for playback. In the later case, a gap may have occurred and the packet is played immediately.

In **video** systems jitter is typically avoided by introducing a frame buffer at the receiver and keeping the jitter to within the boundaries of the size of the frame buffer. Due to the size of storage for the frames usually only two frames are buffered at the receiver. Typically this introduces an additional delay of 80 ms which means a substantial increase of the roundtrip delay. For dialogue applications these 160 ms must be added to all other delays which can often give non-acceptable values. Therefore most communication should be either isochronous or some method of handling the jitter incorporated. Today's research has concentrated on reducing the jitter rather than handling it.

Looking at lost, late or corrupted frames (as LDUs) in a video sequence, we can distinguish three kinds of recovery mechanisms:

- In the most sophisticated case we can try to compensate for the missing frame by presenting them for a longer time (see Figure 11). This is certainly the best way as the viewer will not notice the discontinuity if frames are presented for a fraction longer than the regular frame. However, this method is not of practical value with the current video technology, frame rates are fixed and we can not just adjust frames at will.

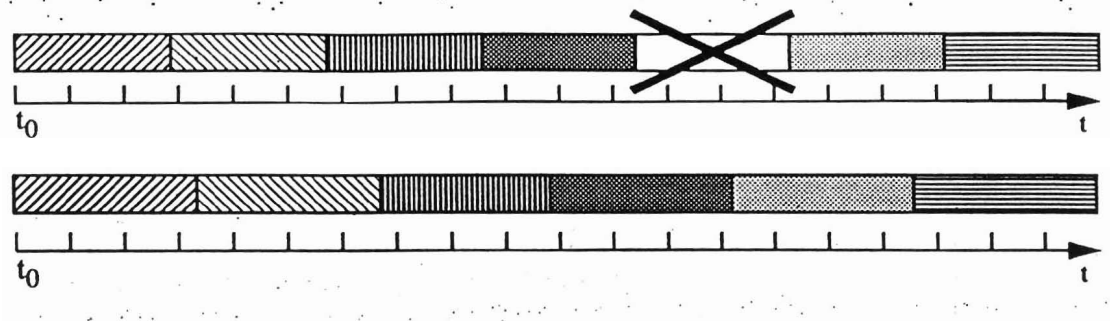

Figure 11: Expanding each frame

- Another technique is that one frame can just be replicated (see Figure 12). This is possible and experiments show this is one reasonable way of compensating for a lost frame.

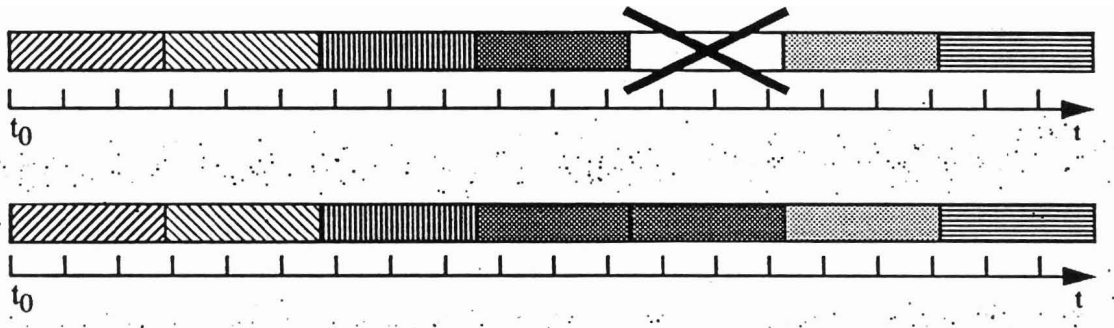

Figure 12: Continuing the data stream by doubling a frame

- The most common method is just to drop the corrupted frame and to continue with the next frame (see Figure 13). Initially this seems an inadequate solution due to the jerkiness of the video, however for a small number of frames we did not found a significant difference from the doubling technique.

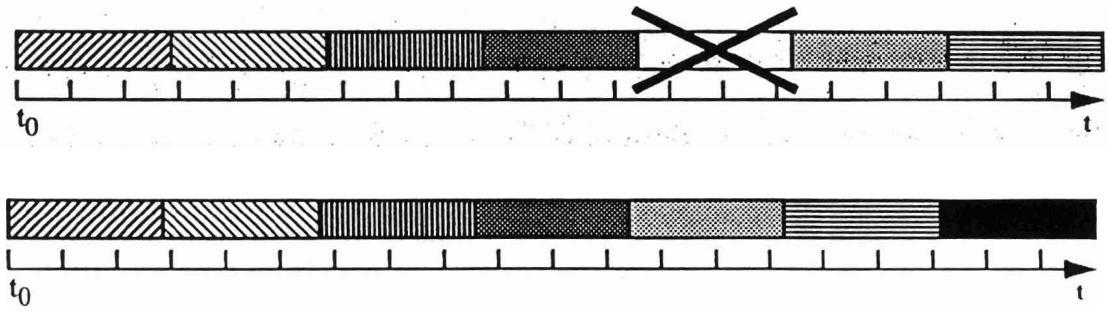

Figure 13: Continuing the data stream without doubling a frame

Jitter can be instantaneously recognized when the video scene contains motion, it can be tolerated at the chance of scenes or if the scene has either very fast or slow movement of objects in front of a static background. At the change of scene we can easily drop up to 15 frames. Between 2 scenes we may introduce up to 3 black frames which will not be noticed by the

viewer. The advances in video parsing makes us believe that we will soon be able to identify changes of scenes in real time [ZKSm93], and we will be able to detect very slow and fast motion in scenes in real time.

Jitter can also be seen in the context of **pointer** synchronization. Jitter of pointer data implies some discontinuity in the display of the pointer at a remote screen, which can certainly more easily be tolerated than jitter of audio or video data.

A pointer is used in CSCW shared window application in two modes:

- The user just wants to show a certain object in the respective window by positioning the pointer on top of this object. Subsequently the user may also push a button in order to perform some operation on this object. In such an application it is important that the viewer easily locates where the pointer is at any given moment. This can best be supported by having pointers with appropriate size, color and shape.
- The pointer is used to show a specific path on the shared window. E.g., the remote pointer is used to describe a route on a map. Another example is to show how a grabbed object is dragged along a certain path and dropped somewhere else on the screen. In any case it should provide the user with the illusion of continuous movement.

In the first case we found out the shortest intervals of how long we typically retain the pointer on some object is about 100ms. Hence 10 pointer updates (with at most 10 changes of pointer location) per second are sufficient for providing the illusion of error-free operation.

The second scenario is more challenging as we need to experience the user feed back for this illusion of continuity and we need to find out how many coordinates we may miss and it will still be seen as continuous movement. For this second scenario initial experiments have shown that not more than 15 pointer updates per second are required.

Knowledge of the skew (without any jitter) provides the means to adjust the buffers and control algorithm at the set-up phase of multimedia data connections. Data communication errors in just one of two synchronized media streams can be handled in a more user friendly way; let us assume that so far there is no skew at the receiver, then a packet at the video channel is corrupted and one frame can not be recovered by the included forward error correction mechanism. In order to make this error less serious we want to keep the audio data continuous, because the following frame is already at the receiver, the playback control algorithm can immediately display this frame with a skew of 40 ms which will not be perceived by the user. Having introduced a non-zero skew, we can reset the skew to be zero without the viewer detecting it at all; i.e. at the end of a video scene or at an audio silence interval.

Tolerable jitter allows for the smoothing of long term changes of rate on the receiver site without any interaction with the sender. Let us assume the clocks of the sender and several receivers are not controlled by a central instance, we may encounter a difference of 33 ms which means that either the receiver buffer may reach a low/high water mark, at this point it would be nice to either introduce/discard one frame. With the notion of jitter perception as described in this section we now know that we can do this, we just need to decide (depending on the content of the video and audio data) where to perform it.

10 Some Final Remarks

In local systems resource management is often easier to provide because there are sufficient resources or it is a single user configuration. In networked systems there are many concurrent processes making use of the same resources, therefore skews between media easily arises. Synchronization QoS parameters allow the builders of distributed multimedia and communication systems to make use of the affordable tolerances.

This paper provides a set of quality of service values for synchronization. It is starting point for media synchronization with extensive user interface experiments, the enforcement of which remains a different topic.

First of all I would like to acknowledge the enthusiastic work done by Clemens Engler; We spent hours and nights of controversial discussions on the expected results, the influencing factors and the design of the experiments, he also carried out most of the experimental work. Martin Engelhardt has started with the detailed evaluation of all jitter related experiments. Wieland Holfelder helped in producing the basic video material, and I would like to acknowledge the patience and accuracy of all our test candidates. Roger Dannenberg, CMU Pittsburgh, provided many valuable hints concerning jitter of audio samples and synchronization related to music. The anonymous reviewers and Ralf Guido Herrtwich provided many valuable comments for the final version of the paper. Thank you.

References

[AnHo91] David P. Anderson, George Homsy: *Synchronization Policies and Mechanisms in a Continuous Media I/O Server*, International Computer Science Institute, Technical Report no. 91-003, Berkeley, CA, 1991.

[BHLM92] Gerold Blakowski, Jens Hübel, Ulrike Langrehr, Max Mühlhäuser: *Tools Support for the Synchronization and Presentation of Distributed Multimedia*, Computer Communications, vol. 15, no. 10, December 1992.

[Bles78] Barry Blesser: *Digitization of Audio: A Comprehensive Examination of Theory, Implementation, and Current Practice*, Journal of the Audio Engineering Society, JAES Vol. 26 no. 10, October 1978, pp. 739-771.

[ClRi94] Mark Claypool, John Riedl: *Silence is Golden? - The Effects of Silence Detection on the CPU Load of an Audio Conference,* Proceedings of the IEEE International Conference on Multimedia Computing and Systems, May 14-19, 1994, Boston, MA, pp. 9-18.

[Clyn85] M. Clynes: *Secrets of Life in Music: Musicality Realized by Computer* in Proceedings of the 1984 International Computer Music Conference, San Francisco, International Computer Music Association, 1985.

[Dann93] Roger Dannenberg: *Sound Effects and Video Synchronization and on Music Playback and Visualization of the Corresponding Strokes,* personal communication, 1993.

[DaSt93] Roger Dannenberg, Richard Stern: *Experiments Concerning the Allowable Skew of Two Audio Channels Operating in the Stereo Mode*, personal communication, 1993.

[DLWe93] Bert J. Dempsey, Joerg Liebeherr, Alfred C. Weaver: *A New Error Control Scheme for Packetized Voice over High-Speed Local Area Networks,* Proceedings of the 18th Conference on Local Computer Networks, Minneapolis, MN, September, 1993.

[Ferr90] Domenico Ferrari: *Client Requirements for Real-Time Communication Services,* IEEE Communications Magazine, November 1990, pp. 65-72.

[Lick51] J.C.R. Licklider: *Basic correlates of the auditory stimulus,* in S. S. Stevens, ed. Handbook of Experimental Psychology, Wiley, 1951.

[LiGh90a] Thomas D.C. Little, A. Ghafoor: *Synchronization and Storage Models for Multimedia Objects,* IEEE Journal on Selected Areas in Communication, vol. 8, no. 3, Apr. 1990, pp. 413-427.

[LiGh90b] Thomas D.C. Little, Arif Ghafoor: *Network Considerations for Distributed Multimedia Objects Composition and Communication,* IEEE Network Magazine, vol. 4 no. 6, November 1990, pp. 32-49.

[LKGe94] Li Li, Achmed Karmouch, Nicolas D. Georganas, *Multimedia Teleorchestra with Independent Sources: Part 1 - Temporal Modeling of Collaborative Multimedia Scenarios, Part 2 - Synchronization Algorithms;* Multimedia Systems, acm/Springer-Verlag, vol.1, no. 4, 1994, pp. 143-165.

[LLKG93] L. Li, L. Lamont, A. Karmouch, N.D. Georganas: *A Distributed Synchronization Control Scheme in A Group-oriented Conferencing Systems,* Proceedings of the second international conference Broadband Islands, Athens, Greece, June 15-16, 1993.

[Nico90] Cosmos Nicolaou: *An Architecture for Real-Time Multimedia Communication Systems,* IEEE Journal on Selected Areas in Communication, vol. 8, no. 3, April 1990, pp. 391-400.

[Ravi92] Kaliappa Ravindran: *Real-time Synchronization of Multimedia Datastreams in High Speed Packet Switching Networks,* Workshop on Multimedia Information Systems (MMIS '92), IEEE Communications Society, Tempe, AZ, February 1992.

[RuAv80] Dean Rubine, Paul McAvinney: *Programmable Finger-tracking Instrument Controllers,* Computer Music Journal, vol. 14, no. 1, Spring 1980, pp. 26-41.

[ShSa90] Doug Shepherd, Michael Salmony: *Extending OSI to Support Synchronization Required by Multimedia Applications,* Computer Communications, vol.13, no.7, September 1990, pp. 399-406.

[Stei90] Ralf Steinmetz: *Synchronization Properties in Multimedia Systems,* IEEE Journal on Selected Areas in Communication, vol. 8, no. 3, April 1990, pp. 401-412.

[Stei92] Ralf Steinmetz: *Multimedia Synchronization Techniques: Experiences Based on Different System Structures,* Proceedings of IEEE Multimedia Workshop '92, Monterey CA, April 1992.

[Stei93] Ralf Steinmetz, *Multimedia-Technology: Introduction and Fundamentals (in German),* Springer-Verlag, September 1993.

[StNa95] Ralf Steinmetz, Klara Nahrstedt: *Fundamentals in Multimedia Computing and Communications,* Prentice-Hall, May 1995.

[Stew87] M. Stewart: *The Feel Factor: Music with Soul,* Electronic Musician, vol. 3, no. 10, 1987, pp. 55-66.

[Stoc72] T. Stockham: *A/D and D/A Converters: Their Effect on Digital Audio Fidelity,* in Digital Signal Processing, L. Rabiner and C. Rader, (Eds.), IEEE Press, NY 1972.

[Wood51] H. Woodrow: *Time Perception,* in S. S. Stevens (Ed.), Handbook of Experimental Psychology, Wiley, 1951.

[ZhKe91] Hui Zhang, Srinivasan Keshav: *Comparison of Rate-Based Service Disciplines,* Proceedings of acm SIGCOMM'91, Zürich, Switzerland, September, 1991.

[ZKSm93] HongJiang Zhang, A. Kankanhalli, Stephen W. Smoliar: *Automatic Parsing of Video,* 'Multimedia Systems', acm/Springer vol. 1, no. 1, pp.10-28.

Appendix A: Detailed Results

In the following, the whole set of results is presented by showing the accumulated answers to the questionnaires. We distinguish between three different views, (1) the 'head view', (2) the 'shoulder view', and (3) the 'body view'.

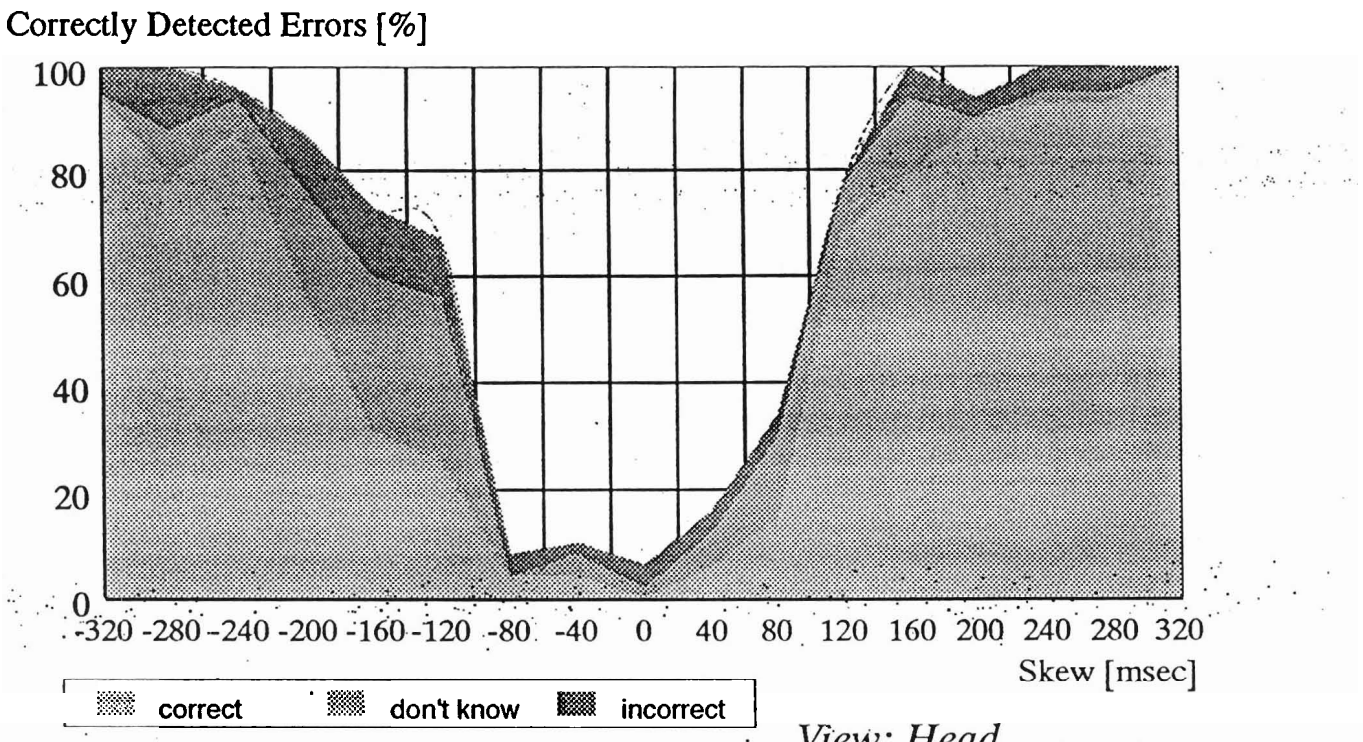

Figure 14: Correct detection of synchronization errors at head view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

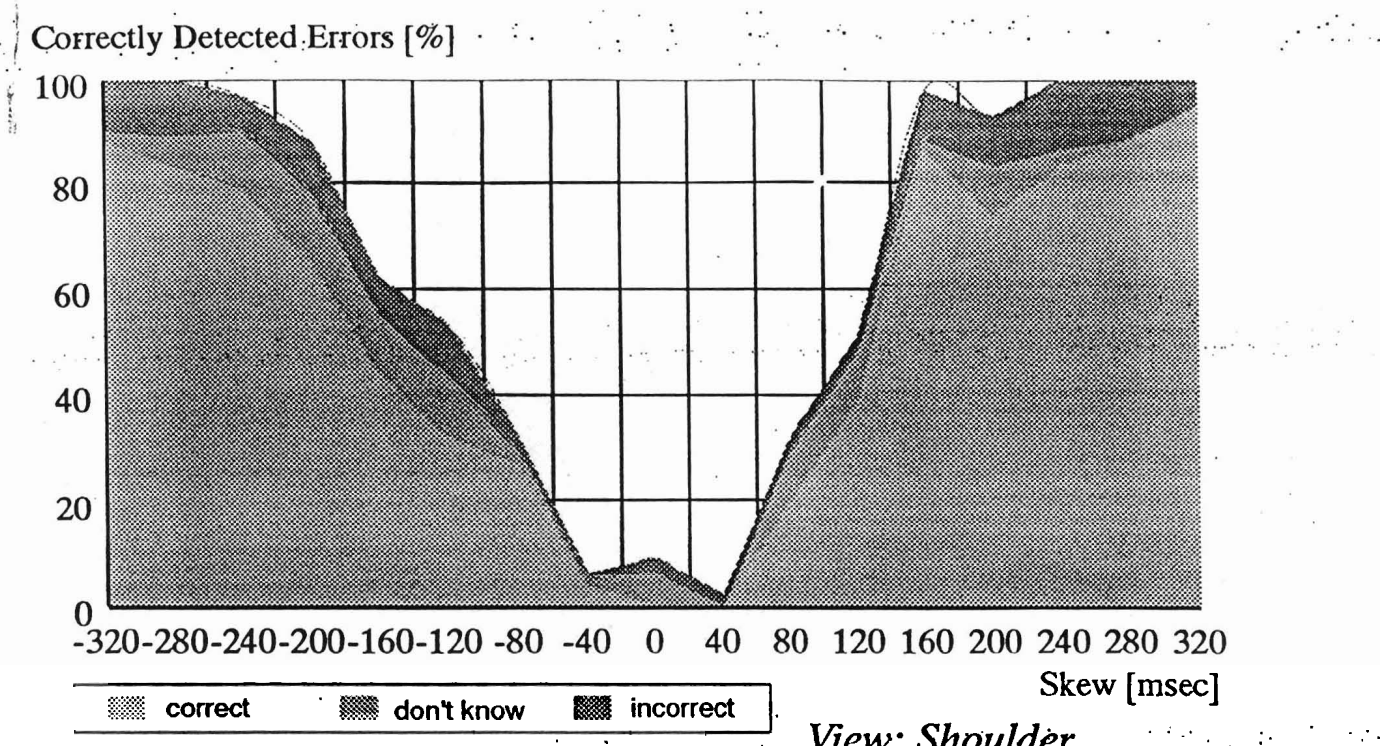

Figure 15: Correct detection of synchronization errors at shoulder view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

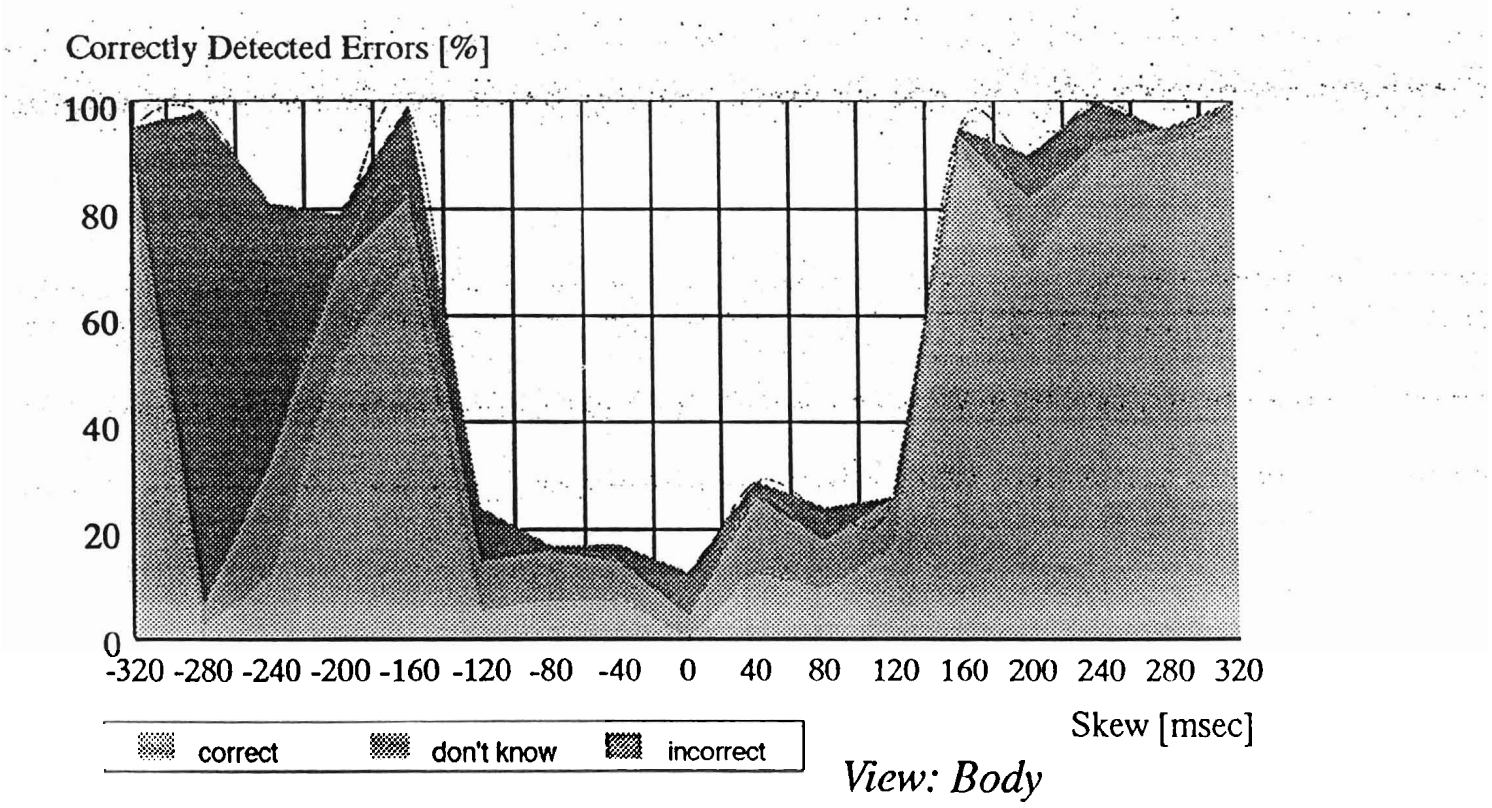

Figure 16: Correct detection of synchronization errors at body view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

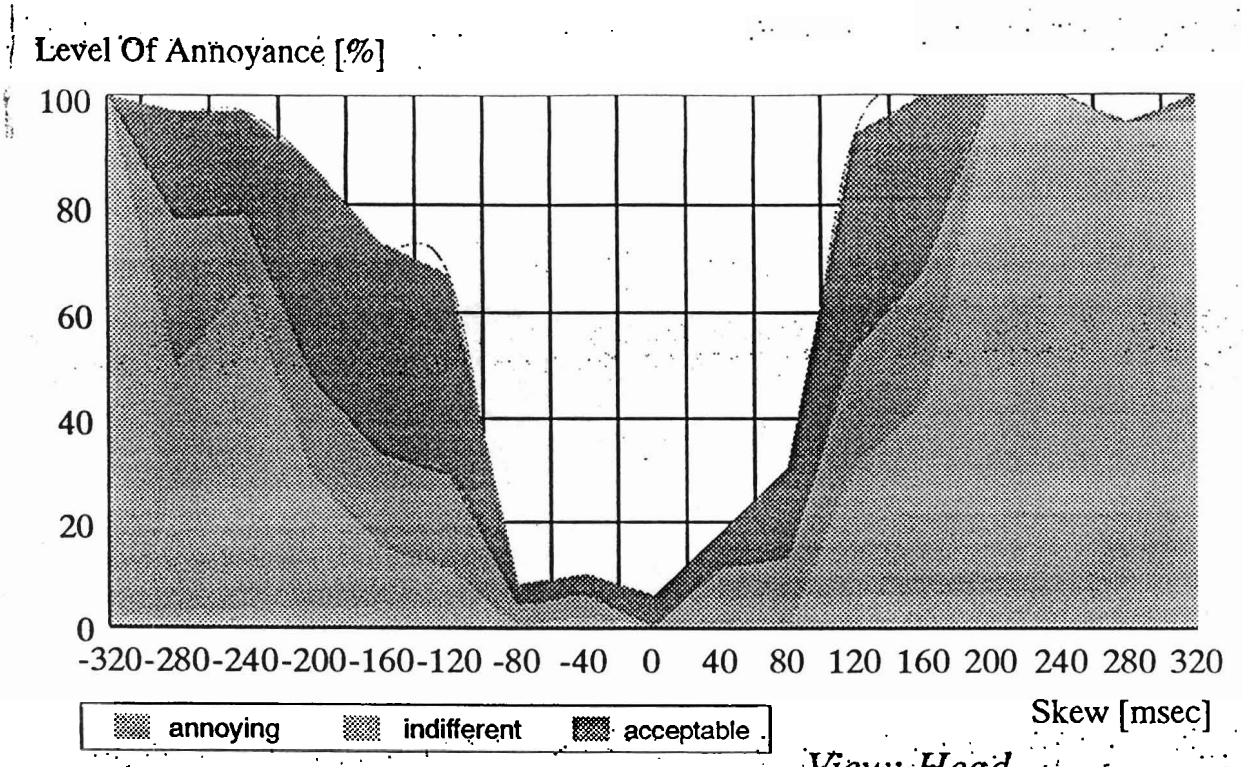

Figure 17: Level of Annoyance at head view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

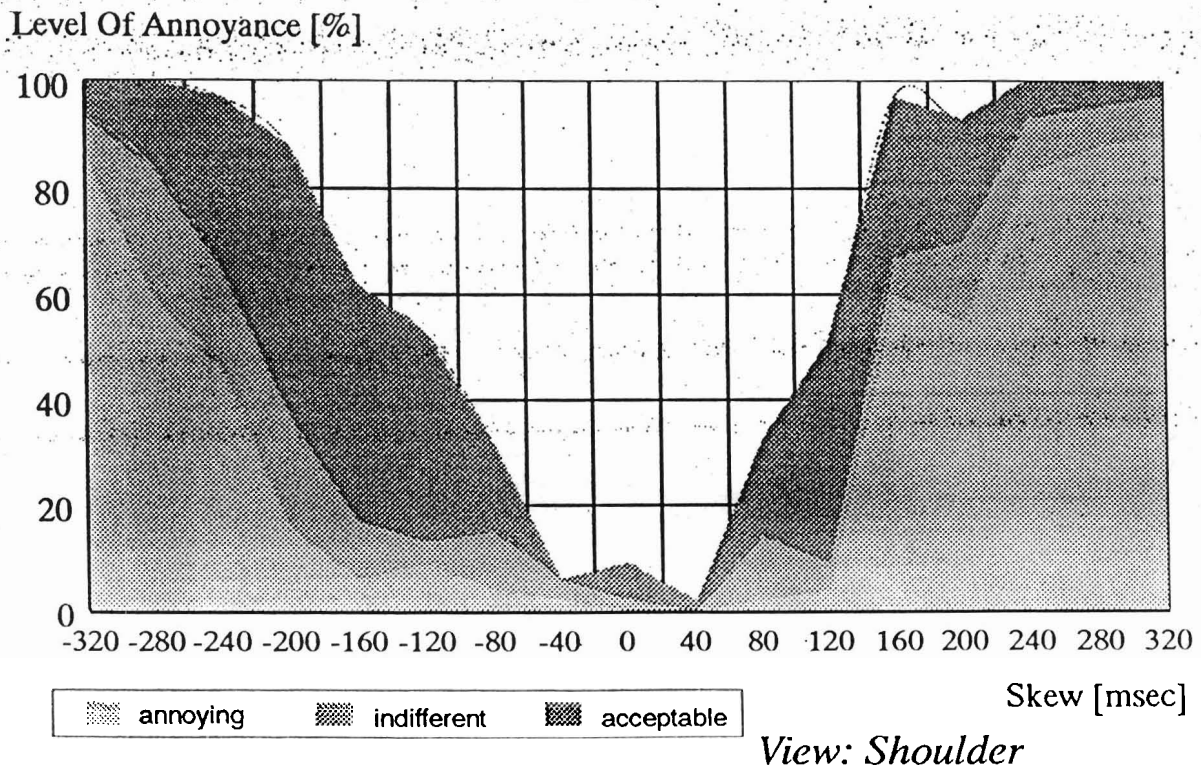

Figure 18: Level of Annoyance at shoulder view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

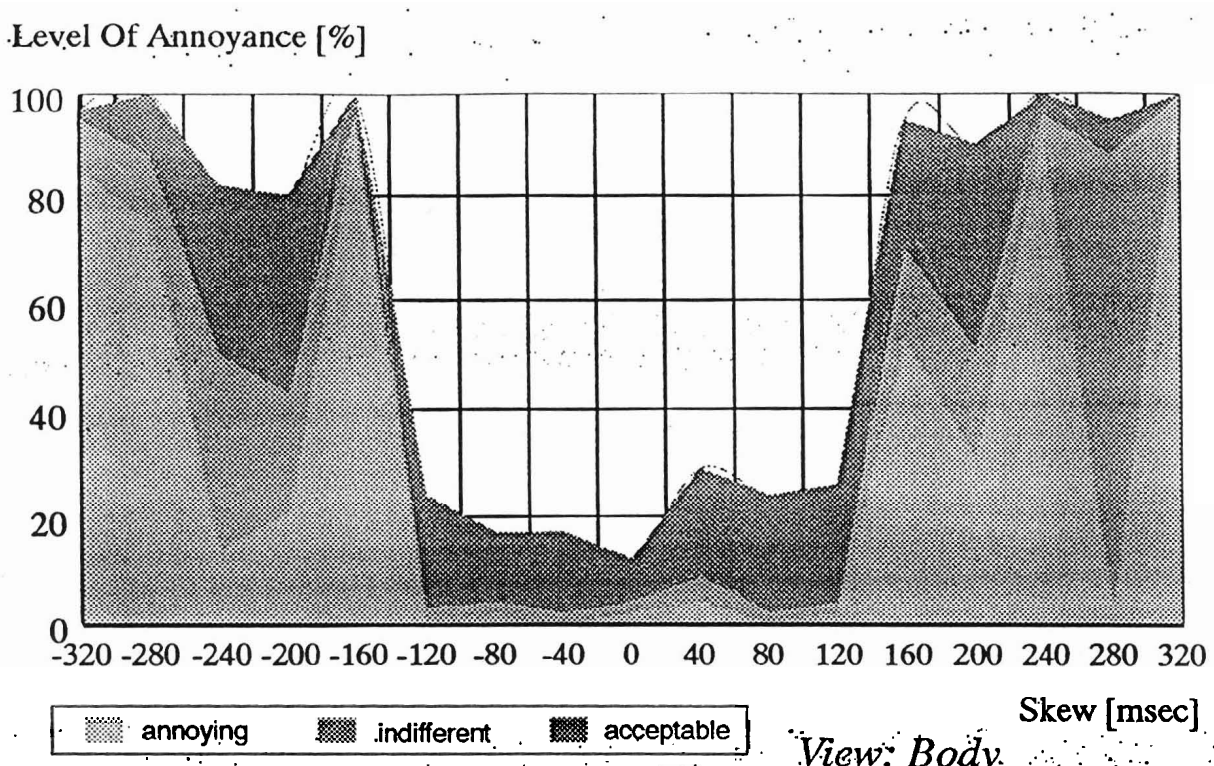

Figure 19: Level of Annoyance at body view

Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

Appendix B: Questionnaire

The questionnaire contained the following set of questions which provided the basis for this analysis. Question 2 and 3 had to be answered on a single choice basis.

① While watching this video clip, did you **detect any artifact** or **strange effect**?

If so, please try to describe it in a few words. (✍)

If you detected a synchronization error please proceed with the following question ②
(otherwise, watch the next clip and proceed with the first question)

②

Are you able to **identify** if audio was ahead of or behind the moving pictures? (✎)

a) Yes, I identify **audio** to be played **ahead of video** ⇨ ❐

b) Yes, I identify **audio** to be played **behind video** ⇨ ❐

c) **No, I notice that audio is out of sync with** respect to video but, I am **not sure** if audio is played ahead of or behind video. ⇨ ❐

Please proceed with question ③

③

You noticed a synchronization error.
How would you **qualify this error** if you have to watch all your TV programs with such an error? (✎)

a) I would not mind, the error is **acceptable** ⇨ ❐

b) I dislike it, the error is **annoying** ⇨ ❐

c) I am **not sure** if I would accept such an error or if I would really dislike it ⇨ ❐

Please proceed to watch the next clip and return to the first question.

Appendix C: Sequencing of Clips

The following Table shows the sequencing of clips as performed in the lip synchronization experiments.

Sequence	Head	Shoulder	Body
1	-80	+160	+120
2	+120	-40	-160
3	+40	-120	-40
4	-200	+240	+160
5	0	-160	-240
6	+80	+280	+80
7	-40	-80	-320
8	+240	-240	0
9	-120	+200	+240
10	+160	+320	-200
11	-240	+40	-120
12	-160	-120	+320
13	+200	-320	-40
14	-320	0	-280
15	-120	-40	+40
16	0	+80	+280
17	-280	-280	-80
18	-40	-200	+200
19	+320	+120	-120

Table 2: Ordering of the Probes