• submitted to: IEEE JSAC
Special Issue on Multimedia Synchronization

• based on changed and
enhanced IBM - Technical Report no. 43 9310, 1993.
"Ralf Steinmetz, Clemens Engler
"Human Perception of Media Synchronization"

[StEn93] *Ralf Steinmetz, Clemens Engler;* **Human Perception of Media Synchronization;** Technical Report 43.9310, IBM European Networking Center Heidelberg, Germany, August 1993; überarbeiteter Auszug erscheint als [Stei94e].

Human Perception of Media Synchronization

Ralf Steinmetz

[StEn93] ***Ralf Steinmetz, Clemens Engler;* Human Perception of Media Synchronization; Technical Report 43.9310, IBM European Networking Center Heidelberg, Germany, August 1993; überarbeiteter Auszug erscheint als [Stei94e].**

IBM European Networking Center
Creative Multimedia Studios
Vangerowstraße 18 • 69115 Heidelberg • Germany

Phone: +49-6221-59-3000 • Fax: +49-6221-59-3400

steinmetz@vnet.ibm.com

Abstract: Multimedia synchronization comprises the definition and the establishment of temporal relationships among audio, video, and other data. The presentation of 'in sync' data streams by computers is essential to achieve a natural impression. If data is 'out of sync', human perception tends to identify the presentation as artificial, strange, or annoying. Therefore, the goal of any multimedia system is to present all data without 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. It leads to a first guideline for the definition of a synchronization quality of service. The results show that a skew between related data streams may still let data appear 'in sync' and it outlines some constraints under which jitter may be tolerated. It also turned out that the notion of a synchronization error highly depends on the types of media. We use our findings to develop a scheme for the processing of non-trivial synchronization skew between more than two data streams.

1 Introduction

We understand multimedia according to [HeSt91b][Stei93][StNa94]; 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 any data and the synchronization between various kinds of media and data are the key issues for integration. Multimedia synchronization is needed to ensure a temporal ordering of events in a multimedia system.

At a first glance this ordering applies to single data streams: a stream consists of consecutive logical data units (LDUs). In the case of an audio stream, LDUs may be individual samples or blocks of samples transferred together from a source to one or more sinks. A video LDU typically corresponds to a single video frame and consecutive LDUs have to be presented at the sink with the same temporal relationship as they were captured at the source leading to intrastream synchronization.

The temporal ordering also applies between related data streams. The most often discussed relationship is the simultaneous playback of audio and video with 'lip synchronization'. Both kinds of media must be 'in sync', otherwise the viewer would not be satisfied with the presentation. In general an interstream synchronization involves relationships between all kind of media including pointers, graphics/images, animation, text, audio, and video. In the following, 'synchronization' always means interstream synchronization.

For delivering multimedia data correctly at the user interface, synchronization is essential. Unlike other notions of correctness, it is not possible to provide an objective measurement for synchronization. As human perception varies from person to person, only heuristic criteria can determine whether a stream presentation is correct or not. This paper presents our results of some extensive experiments related to human perception of synchronization between different media.

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) and, hence, synchronized at the source and demultiplexed just before presentation at the sink. Multiplexing is not always possible and wanted, e.g. because multimedia data needs to go through different routes in a computing system. The separate handling of previously related data leads to time lags between the media streams. These lags have to be adjusted at the sink for 'in sync' presentation.

Some work on how to implement multimedia synchronization was done in related projects [AnHo91] [Blak92] [LKGe92] [LLKG93] [ShSa90] [Stei92]. Work has also been devoted to define synchronization requirements [LiGh90] [LiGh90b] [Nico90] [Ravi92] [Stei90]. It is often reported that audio can be played up to 120 ms ahead of video and in the reverse situation video can be displayed 240 ms ahead of audio. Both temporal skews will sometimes be noticed, but can easily be tolerated without any inconvenience by the user [Murp90]. Some authors report a skew of +/-16 ms [Dann93] or no skew at all to be acceptable.

Implementing our own synchronization mechanisms, we were unable to draw the right conclusions from these reports - their statements were contradictory. There was a lack of an in-depth analysis of synchronization between the various kind of media and, in particular, for lip and pointer synchronization. We decided to conduct our own study and to explore these fundamental issues 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 main results of lip synchronization experiments, the notion of the 'quality of synchronization' is elaborated in

Section 3. Section 4 describes the test strategy, how the results were achieved including influencing factors. Section 5 presents the results on pointer synchronization, remaining types of media synchronization are discussed in Section 6. The aggregation of various individual media synchronization results is analyzed in Section 7 and Section 8 defines and summarizes the results in terms of the required quality of service parameters. In Section 9 first results of human perception of jitter are described. The appendix of this paper includes an example of the questionnaire used by test participants and shows all results in form of appropriate graphics.

2 The Lip Synchronization Experiment

'Lip synchronization' denotes the temporal relationship between an audio and a video stream where speakers are shown while they say something. 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 which skews were perceived as 'out of sync' for audio and video data. 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 several discussions with experts working with audio and video, we noticed that most of the personal experiences were derived from very specific situations. As an immediate consequence we have been confronted with a wide range and tolerance levels up to 240 ms. A comparison and a general usage of these values is doubtful because the environments from which they resulted were were comperable. In some cases we encountered the 'head view' displayed in front of some single color background on a high resolution professional monitor. In another set-up a 'body view' was displayed in a video window at a resolution of 240*256 pixels in the middle of some dancing people. In order to get the most accurate and stringent affordable skew tolerance levels, we selected a speaker in a TV news environment as a 'talking head' (see Figure 1). In this scenario, the viewer is not disturbed by background information. The user is attracted by the gestures, eyes, and lip movement of the speaker. We selected a speaker who makes use of gestures and articulates very accurately.

We recorded the presentation and then played it back in our experiments with artificially introduced skew that was adjusted according to the frame rate, i.e., n times 40 ms, that was introduced by professional video editing equipment. We conducted some experiments with a higher resolution time scale by cutting the material with the help of a computer where it was possible to introduce a smaller delay in the audio stream. It turned out that there was no need for any test with higher granularity than 40 ms.

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

We expected a relationship between the detectable skew and the actual size of the head displayed at the monitor. As shown in Figure 1 we selected three different views of the speaker. At the very close 'head view' the head completely fills the screen, the 'shoulder view' shows the head as well as the shoulders while the third 'body view' captures the whole person sitting in a room.

Lip synchronization usually applies to speech as an acoustic signal related to its visual representation of the speaker. We expand this notion to cover the correlation between noise and its visual appearance, e.g., clapping. For the latter, our experiments included a person working with a hammer and some nails. Nevertheless the most exhaustive study was performed in the news environment.

Figure 2 provides an overview of the main results. The vertical axis denotes the relative amount of test candidates who detected a synchronization error, regardless of being able to determine if audio was before or after the video. As one might expect, if the skew is relatively small most of the people did not notice it; large skews became obvious. However, 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.

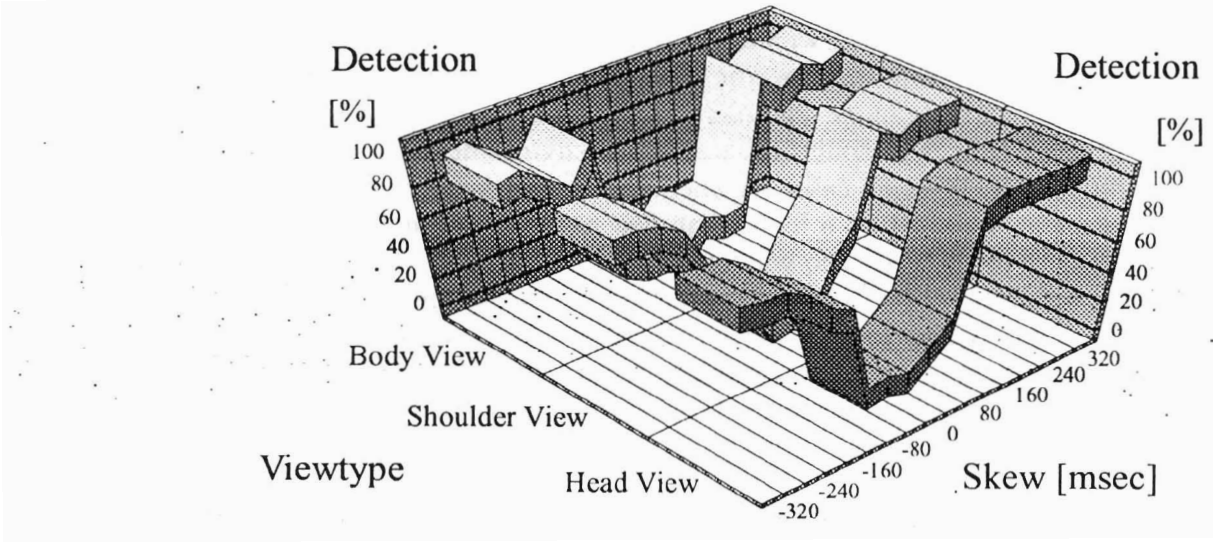

Figure 2: Detection of synchronization errors with respect to the three different views. Left part, negative skew; video ahead of audio; right part, positive skew; video behind audio

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.

1. Here we just outline the different views, the quality of the original clips is TV-like.

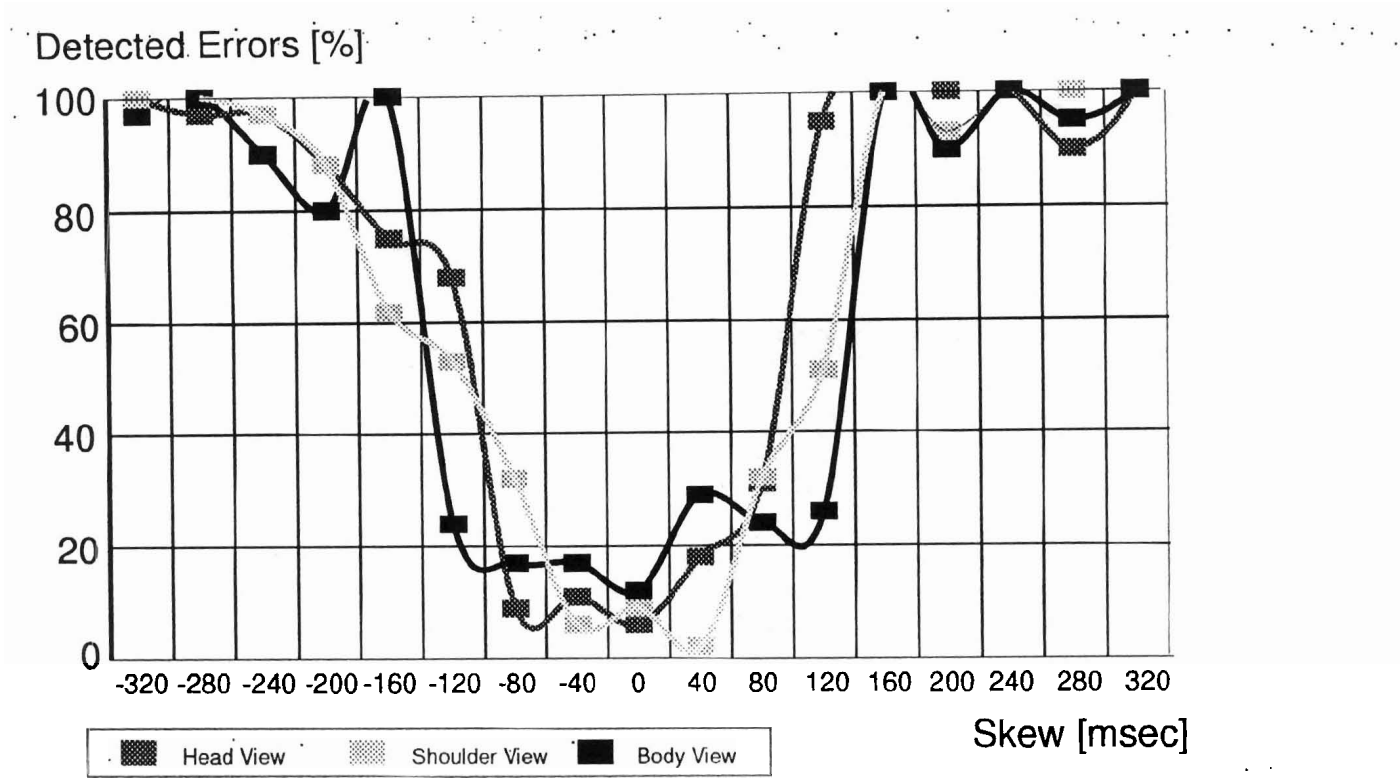

Figure 3: Detection of synchronization errors
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

The left side of the figure relates to negative skew values, where video is ahead of audio. In our daily life, we experience this situation whenever we talk to some distant located person. All three curves are, in general, flat in this region. Since we are not accustomed to hearing speech ahead of the related visual impression, the right side of the curves turns out to be steeper.

The 'body view' curve is broader than the 'head view' curve, at the 'head view' a small skew was easier to notice. This was more difficult in the 'body view'. The 'head view' is also more asymmetric than the 'body view'. Basically, the further away we are situated, the less noticeable the error is.

At a fairly high skew, the curves show some periodic ripples. This is more obvious in the case of audio being ahead of video. It means that some people had difficulties in identifying the synchronization error even with fairly high skew values. A careful analysis of this phenomenon lead to the following explanation; 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, at the end, at the borders of pauses, and whenever changing drastically the mood (e.g., from an explanation style to a sudden aggressive comment). Errors were more difficult to notice in the middle of sentences. A subsequent test containing video clips with skews according to these minima (without pauses and not showing the start, the end, and changes of mood) caused problems in identifying if there was indeed a synchronization error.

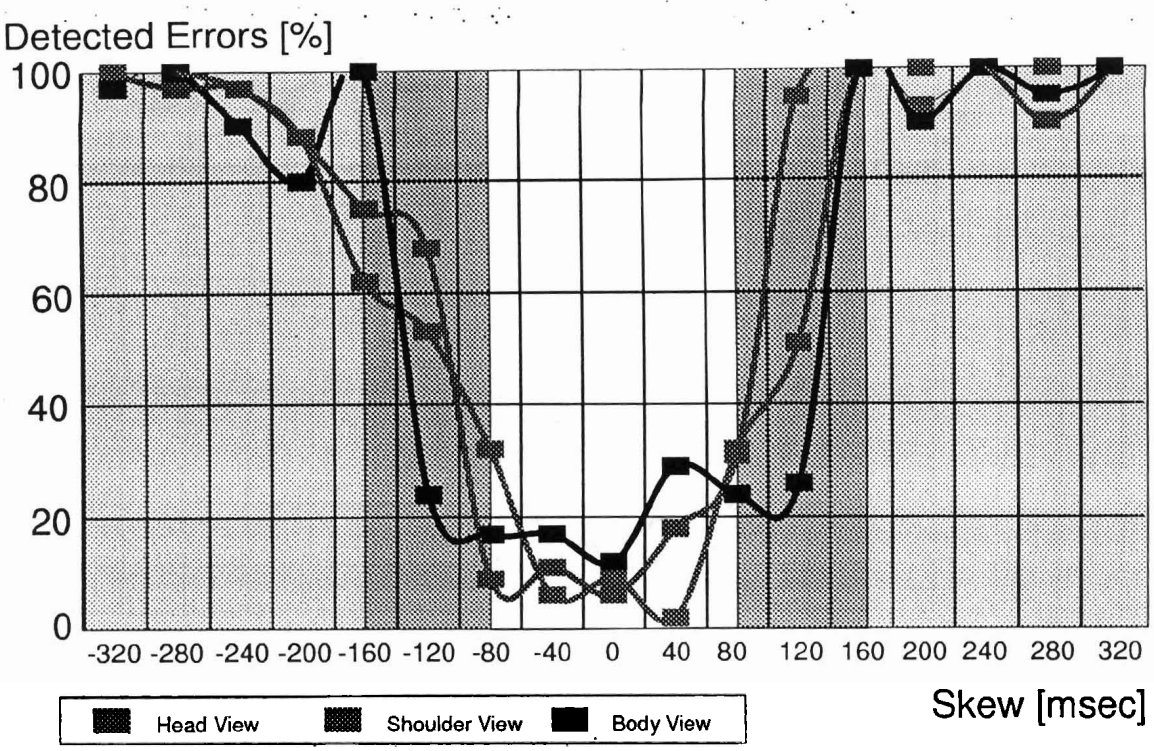

Figure 4: Areas related to the detection of synchronization errors
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

The main results of about 100 test participants are captured in Figure 4 which is composed of different areas:

The **'in sync' area** spans a skew of between -80 ms (audio after video) and +80 ms (audio ahead of video). In this zone most of the users did not detect the synchronization error. Very few mention that if there is an error it does affect their notion of quality video. Additionally, we had some results where test candidates mention that the perfect 'in sync' clip (skew = 0ms) is 'out of sync'. Therefore, we introduced a range of uncertainty in the graph which captures these types of inconsistencies. We came to realize that lip synchronization tolerates the above mentioned skew, this result applies to any type of lip synchronization.

- 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 is, the easier errors are detected and described it as disturbing. The same applies to the overall resolution; the better the resolution is, 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 the other transient area as long as the skew values are near the in sync area. The closer the speaker is, the more obvious the skew is. Apart from this effect we noticed that video ahead of audio can easier be tolerated than the vice versa.

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 to the previous one.

Concerning the different areas, we got similar results with the noise and video experiment (hammer with nails) although the transient areas are more narrow. In this experiment, the type of view had a negligeable influence. The presentation of some violinist in a concert and a choir did not show more stringent skew demands than the speaker being synchronized.

A comparison between sets of experiments ran in English and German 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 usually watched.

Professionals (cutters and TV related technical personnel) showed a smaller level of skew tolerance. If they detected an error, they could correctly state if audio is ahead of or behind video. With the used TV quality a skew of 40 ms was very rarely noticed, the 80 ms skew was sometimes detected. A discussion with professional video cutting teams showed similar results. One out of three professionals stated that she/he would recognize an error with 40 ms of skew, all mentioned that they would recognize a 'lip sync error' starting at 80 ms 'out of sync', but that this might not influence the quality of the perceived information.

3 Quality of Lip Synchronization

Figure 3 and Figure 4 outline the perception of synchronization errors. More important than just to notice the error is the effect of such an 'out of sync' video clip on the human perception. If in an extreme case all people tend to like audio data to be, e.g., 40 ms ahead of video, we should take it into account. 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' which quantifies the quality of synchronization.

Figure 5 shows by which degree a skew was believed to be acceptable or intolerable. We used the 'talking head' experiment and depict here the 'shoulder view' as it is a compromise between the 'head' and the 'body view'. The diagrams of the other views are included in the appendix.

The envelope curve defines the amount of candidates who detected a synchronization problem, i.e., if candidates did not notice an error, they can hardly determine if the error is acceptable or not. This is the same curve for the 'shoulder view' as shown in Figure 4 and Figure 5 without a spline interpolation.

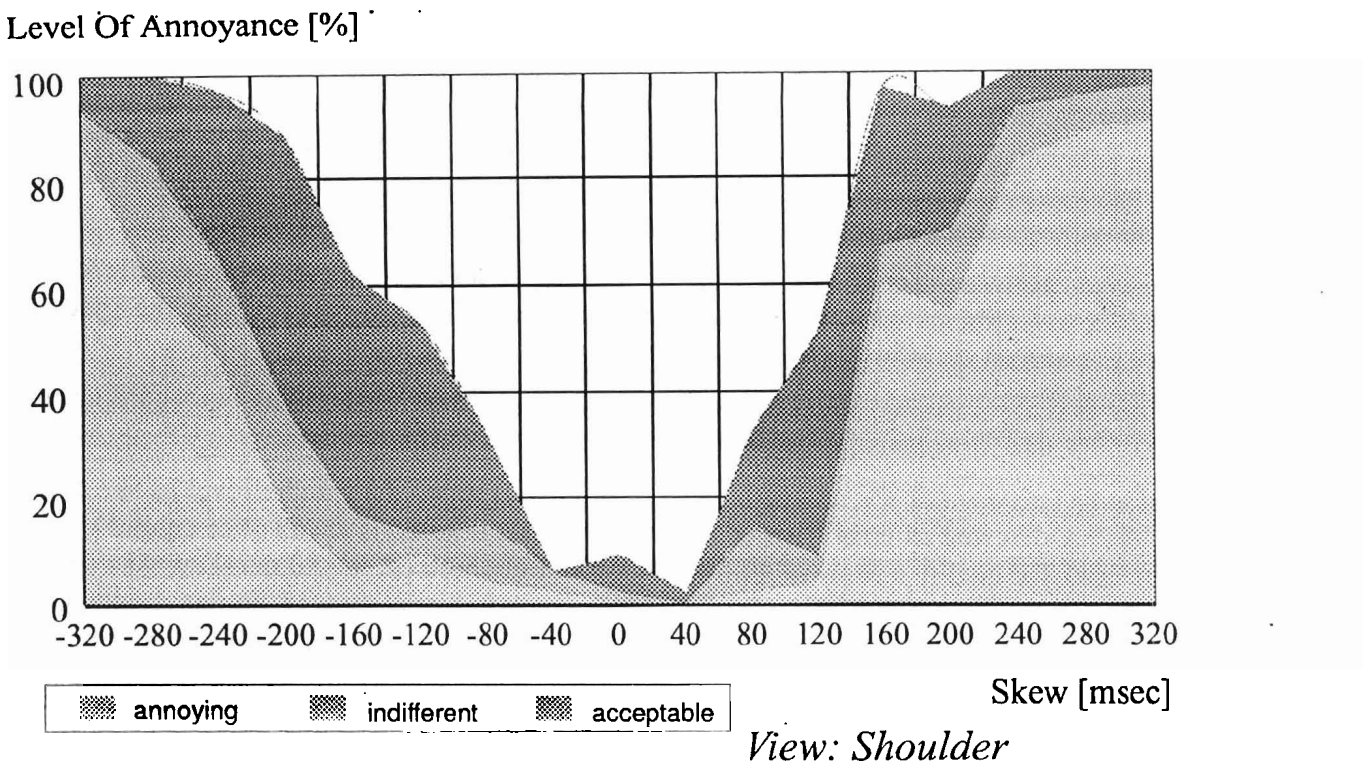

Figure 5: Level of annoyance at shoulder view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

The dark grey areas relate to all test candidates who would accept to listen to and watch video with this synchronization error. In a small follow-on experiment we selected a few test candidates who would tolerate such a skew and presented them a whole movie with this synchronization error. We chose a skew of -160 ms (video ahead of audio). They did not complain at all and very soon concentrated on the content instead of being attracted by looking for some synchronization problem. The curve at the bottom of the dark grey area shows an obvious asymmetry which occurs due to the more natural acceptance of the visual perception being ahead of the related acoustic impression.

The light grey area relates to all people who really dislike this skew and were distracted by it. It also contains the asymmetry discussed above. During the evaluation phase of this study on synchronization, we introduced a skew of +80 ms and -80 ms into two whole movies. These movies were shown to a few candidates who mentioned that such a skew is annoying. It turned out that after a short discussion if we really introduced this artifact (or if we cheat), they did not object at all. The same experiment 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.

This evaluation of the level of annoyance provides a further argument for allowing the skew of lip synchronization to take values between -80 ms and +80 ms as mentioned in the former section.

4 Test Strategy

For each person, the lip synchronization test took approximately 45 minutes. The experiment was intentionally carried out with the same audio and video over and over again. This led to some concentration problems during the whole test, which was alleviated by introducing breaks.

We always ran all tests related to one view in one session. Then, the second and the third view were shown in their sessions. The order of the sessions had no effect. Individual probes, each having a different skew, were shown randomly. This led to sequences of probes as summed up in Table 2 in the Appendix.

Initial experiments showed that a total length of about 30s with a small subsequent break is sufficient for getting the users impression. All experiments with longer video clips did not provide any additional new or different results. With some test candidates, were more experienced with video technology and synchronization issues, 5s turned out to be sufficient. Nevertheless we sticked to have 30s for each probe.

The background of all scenes was static (i.e., not moving) and out of focus in order to keep the distraction to a minimum. In short clips with a moving background the viewer is sometimes more attracted by the actions occurring behind the speaker than by the speaker himself. This would lead to more larger skew values for the perception of synchronization errors. We focussed on the detection of such errors in the most challenging set-ups, this allowed the determination of skew values independently from the actual content of the video and audio data. In these experiments the viewer should never have been distracted by the background.

The same consideration, i.e. background vs. foreground, can be applied to the audio data. The voice of the speaker can be mixed with some background noise or music. In order to differentiate between foreground and background, the volume of the speaker should be at least twice the volume of the background audio. In contrast to the video analogy discussed in the previous paragraph, any background audio did not influence our results. Background noise in the audio channel had no effect on the experiments.

The group of people was selected according to an equal distribution of sex and ages. To have a representative distribution we did not take into account habits (like the time spent for watching TV) and the social status or any other characteristics of the test candidates.

It would have been very interesting if, before presenting each probe to the candidates, they were not aware of the fact that we were looking for synchronization issues. As soon as the test candidates noticed the first time a synchronization fault, they would not have been allowed to continue the experiment with further skews. This would have led to results for casual unexperienced users. As a matter of fact, we started to run the experiment in this way with a very few people. It turned out, that lip synchronization is not detected so easily leading to a broader range of the 'in sync' zone. In order to provide results for building multimedia systems for all types of users, we have to make the assumption that a user can also 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.

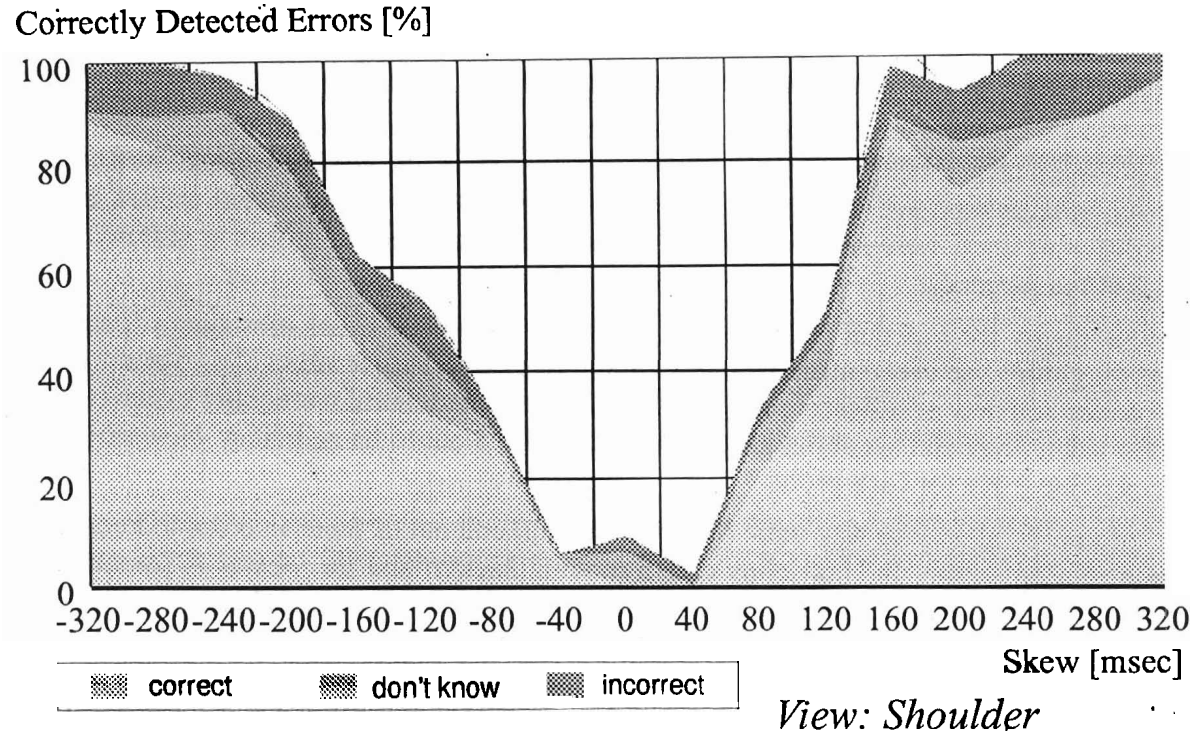

Figure 6: Correct detection of the perceived synchronization error
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

For the purpose of double checking, the candidates were asked to define exactly which type of synchronization error they noticed. It is easier just to detect that something is wrong than to precisely 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.

The envelope curve of Figure 6 defines the amount of candidates who detected a synchronization problem. This is 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 amount of people who detected a mismatch of audio and video.

Nearby the error-free synchronization value (at 0 ms) it was difficult to determine the type of skew, as soon as we had values beyond -40 ms or above +40 ms almost everybody provided correct answers.

5 The Pointer Synchronization Experiment

In a computer-supported cooperative work (CSCW) environment, cameras and microphones are attached to the users' workstations. The audio and video data of one participant is simultaneously presented at the other remote workstation(s). In our next experiment, we assumed the issue of the discussion is a business report including data related to some graphics. All participants have a window with these graphics on their desktop where a shared telepointer is used in the discussion. With the pointer, speakers point to individual elements of the graphics which they are referring to while speaking simultaneously. This requires synchronization of audio and the remote telepointer.

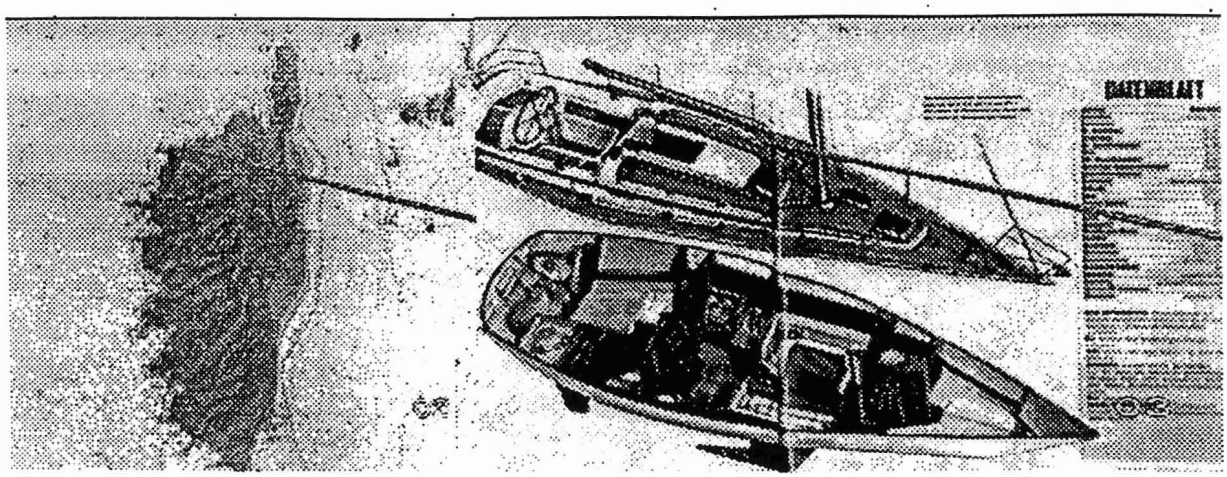

Figure 7: Pointer synchronization experiment based on a map and on a technical sketch

We generated two experiments:

- In the first experiment some technical items of a sailing boat are explained while a pointer locates the aread of interest (see Figure 7, right side). The shorter the explanation, the more crucial the synchronization turns out to be.
- Therefore we additionally made a second experiment with the explanation of a travelling route on a map as seen on the left side of Figure 7.

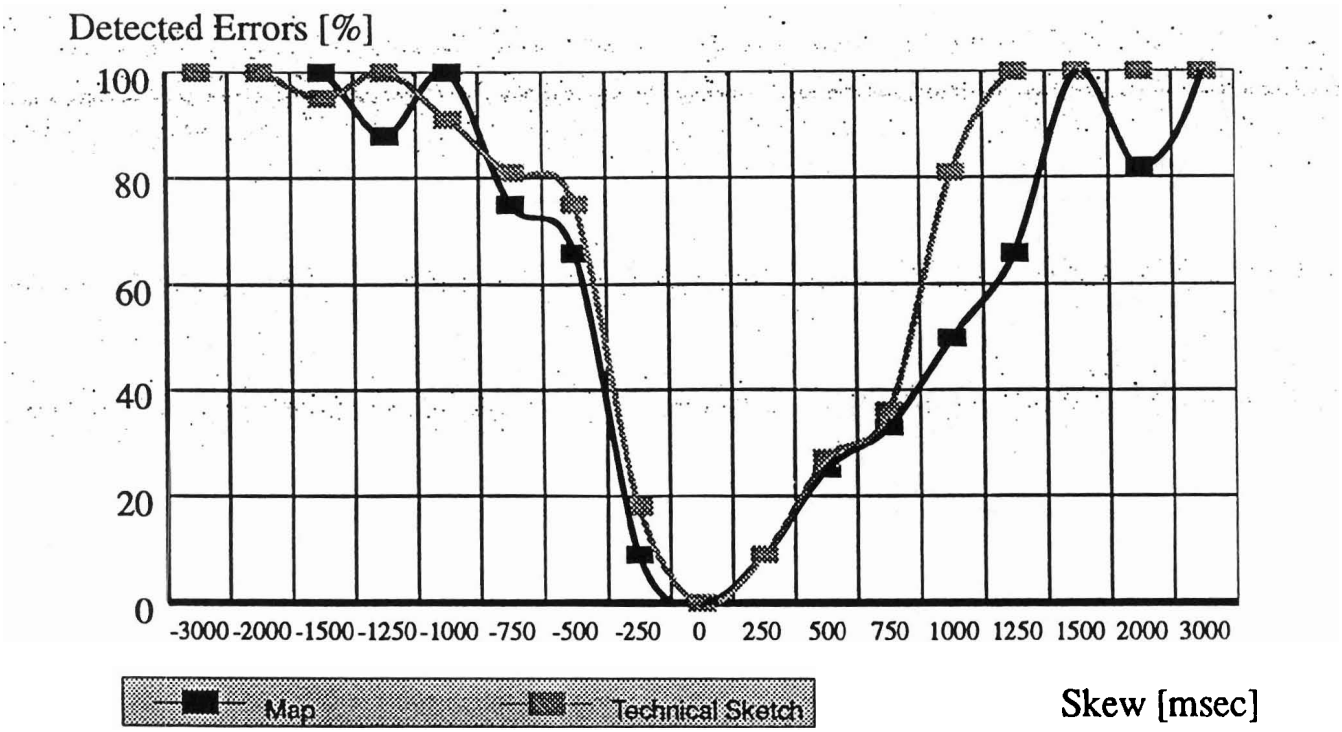

Figure 8: Detection of the pointer synchronization errors
Left part, negative skew; pointer ahead of audio; right part, positive skew; pointer behind audio

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

Using the same margins as in our first experiments, the **'in sync' area** related to audio ahead of pointing is 750 ms and for pointing ahead of audio it is 500 ms. In most of the daily occurring discussions using a telepointer these results can be relaxed. 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 application obsolete, they are distracted and the speaker has to slow down the explanation or carefully move the pointer. From the user interface perspective, this is not acceptable. The experiment of pointing to different locations on the technical figure turned out to be more challenging than the continuous pointer could move on the map. Therefore, the values of the edges of the 'out of sync' area are derived from the pointing on the technical drawing.

In the **'transient' area** we experienced that many test candidates noticed the 'out of sync' effect, but it was not mentioned to be annoying. This is certainly different to 'lip sync' where the user is more sensitive to the faults allowing for less skew and is immediately annoyed.

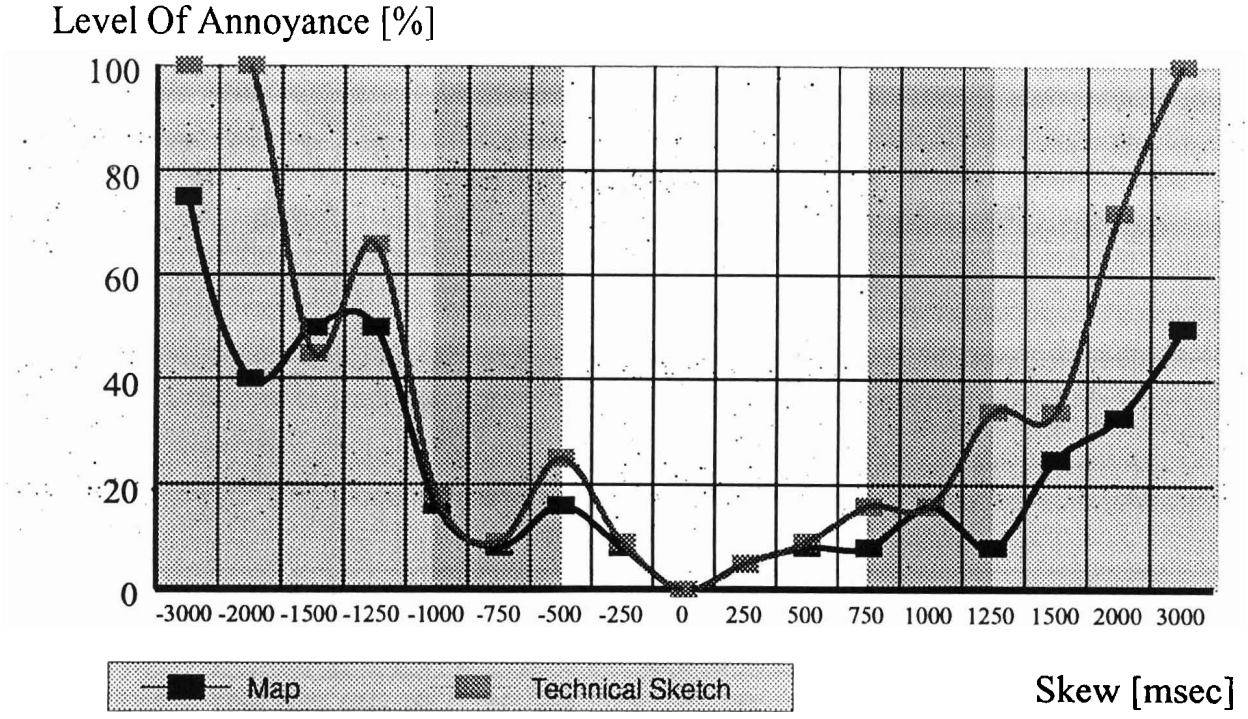

Figure 9: Level of Annoyance of the pointer synchronization errors
Left part, negative skew: pointer ahead of audio; right part, positive skew: pointer behind audio

In Figure 9 these areas are included as part of the diagram of the 'level of annoyance'. These curves denote the relative amount of people who dislike or are indifferent towards the pointer synchronization error. It is remarkable that for several skew values most of the test candidates

detected the fault, but did not object at all to work with such a skew. Therefore, we encounter a broad "in sync' and 'transient' area.

6 Elementary Media Synchronization

Lip synchronization and pointer synchronisation were investigated by us because of the contradictory results available to us from other sources. In the following, we will summrize other synchronization results which we found to provide less diverse statements. We do this to arrive at a more complete picture of synchronization requirements.

Since the beginnings of digital *audio* the ('affordable') *jitter* and the jitter to be tolerated by dedicated hardware has been studied. In discussions with Dannenberg, he provided us some references and the following explanations of these studies: In [Bles78] the maximum allowable jitter in a sample period for at 16 bit quality audio is mentioned to be 200ps, this is explained as the error equivalence to the magnitude of the LSB (least significant bit) of a full-level maximum-frequency 20KHz signal. In [Stoc72] some perception experiments recommend 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 a single percept was mentioned to be 2ms (as cited by [RuAv80]).

The combination of *audio and animation* is usually not as stringent as lip synchronization. A multimedia course on dancing, e.g., comprises the dancing steps as animation with the respective music. By making use of the interactive capabilities, individual sequences can be viewed and listened to over and over again. In this example the synchronization between music and animation is particulary important. Experience showed that a skew of +/- 80 ms fulfills all user demands even though some jitters may occur. Nevertheless, the most challenging issue is the correlation between a noisy event and its visual representation, e.g. a simulated crash of 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 a sound's source location 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 less than total delay. A 2 ms number refers to the synchronization between the onset times of two nominally simultaneous notes or the timing accuracy of notes in sequence, see also [Clyn85] [RuAv80] [Stew87].

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 is 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 overlayed 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.

Sometimes *video* is combined *with animation* as there may be a film where some actors become animated pictures. But, for the following short reasoning of synchronization between video and animation let us go back to the example of a video showing the stroke of a billiard ball and the image of the actual 'route' of this ball. Instead of the static image, the track of the ball can be followed by an animation which displays this route at the time the ball is moving on 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 slidely different position: a good result can be achieved, if in every 3 subsequent video frames, the ball moves by the distance of it's diameter. Less frames will result in the problem of visibility of what occurs, e.g., in tennis, and it may lead to difficulties with the notion of continuity. Derived from this number of 3 subsequent frames, we allow the equivalent skew of 120 ms to occur. This is very tight synchronization, and we have not found any practical requirement which cannot be handled with this value of the affordable skew.

Multimedia systems also incorporate the real-time processing of *control data* and the presentation of this data using various media. A tight timing requirement occurs if the person has to react to this displayed data. 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 two 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. Take as an example an ongoing conference where a video window displays the actual speaker, the audio data is his/her voice as he/she explains some technical details of a new space command station.

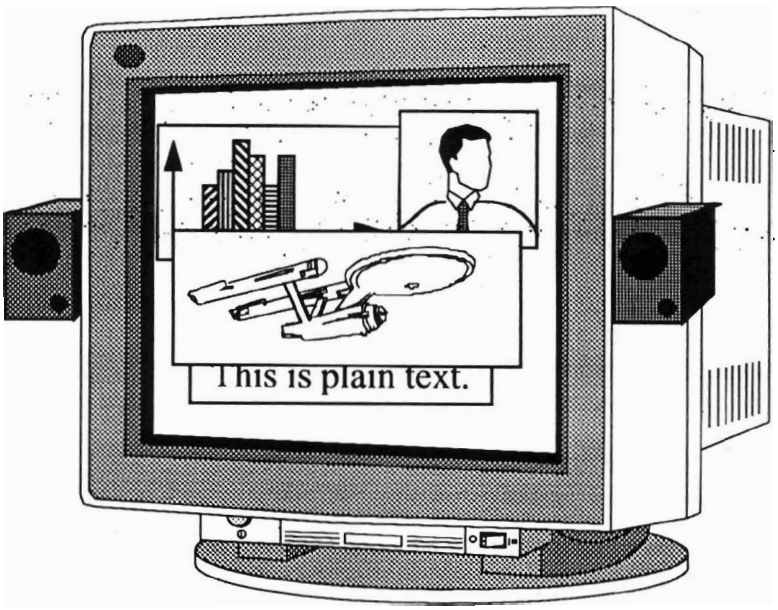

Figure 10: Aggregation of media at the user interface

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 transitive 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. The additional skew is linear dependent with respect to the already provided canonical skew relationships.

In some cases exist too many specifications of a synchronization skew: let us picture a lesson for learning a language that consists of 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. Additionally the sentences need to be synchronized in order to be able to switch on this basis from one language to the other. 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 where all of them need to be fulfilled. We have to find 'the greatest common denominator'. Therefore, in the first step for each available linear independent canonical form all 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 requirements is 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

In the following step any set of linear independent synchronization requirements can be chosen to be used as it may be the following set.

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 sketched procedures allow 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 at 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. The result of this study is of service to this management component. Synchronization requirements can be expressed by a quality of service (QoS) definition. This QoS parameter defines the acceptable skew within the involved data streams, 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 the video data is to be presented simultaneously to some audio and, both are stored as different files or as different entries in a database, lip synchronization according to the above mentioned results has to 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 a subsequent playback. The stored data should be captured and recorded with no skew at all, i.e. it is achieved totally "in sync". This is of particular interest if the file is stored in an interleaved format applying multiplexing techniques. Imagine a participant of an audio-video conference who additionally records this audiovisual data to be playbacked later for a remote spectator. At the conference participant's site, the actual incoming audiovisual data is 'in sync' according to the defined lip synchronization boundaries. Let the data arrive with a skew of +80 ms and let audio and video LDUs be transmitted as a single multiplexed stream over the same transport connection. It is displayed to the user and directly stored on the harddisk (still having this skew). Later on, this data is presented simultaneously at a local workstation and to the remote spectator. For a correct data to be deliverable, 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 - it might turn out 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 whole set of 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. Therefore, in advanced systems, data may also be recorded 'out of sync' leading to an 'in sync' presentation.

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 participiants)	+/- 120 ms
		loosly coupled (e.g. background music)	+/- 500 ms
	image	tightly couppled (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 the literature as referenced in the paper. To our understanding, they serve as a general guideline for any QoS specification. During the lip and pointer synchronization experiments we learned that there are many factors such as the distance of a speaker which to some extend influence these result. We understand that this whole set of QoS parameters as first order result to serve as a general guidance. These values may be relaxed using the knowledge on the actual content.

9 Perception of Jitter

So far we have always 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 at the sender and the same packet being received at the receiver site; this is known to be the end-to-end delay. In asynchronous networks this delay varies. Jitter is defined to be the maximum difference between end-to-end delays experienced by any two consecutive packets [ZhKe91]. Hence jitter implies a varying packet (and LDU) rate at the receiver. This notion can be easily adapted to the human perception environment: Jitter can either introduce gaps in the continuous playback of a data streams (it interrupts this playback) or it shortens the playback of some LDU (a group of audio samples or a video frame).

Until now all multimedia systems try to avoid any jitter in audio and video data streams; all mechanisms are conceived for a complete error-free continuous media data presentation at the user interface. However, the user does not perceive all errors to be very serious and he/she may even not perceive some at all. Therefore, we looked at what a user really perceives as being an error-free data presentation while the presentation itself contains some kind of temporal error.

Jitter in packetized **audio** transmission is commonly addressed by buffering at the presentation site, i.e., at the receiver. 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, all our experiments showed that glitches are immediately detected by any listener if audio or voice is played back at that specific moment. However, voice data is known to consist of talkspurts and silence periods. 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. There the slack time of a packet is defined as the time difference between its arrival time at the receiver and its playback time [DLW93], which 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 within the boundaries of the size of the frame buffer. Due to the waste of storage for the establishment of a jitter-free video playback based on asynchronous communications in most practical implementations at most two frames are buffered at the receiver. With the European 25 fps this introduces an additional delay of 80 ms which mean a substantial increase of the roundtrip delay. For dialogue applications these 160 ms must be added to all other delays and finally it results in non-acceptable values. Hence, either most communication shall be isochronous or we need to be able to cope with some jitter. Until today all approaches tried to avoid jitter at all. However, such a video jitter is not always perceived by the humans as being considered faulty.

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

- In the first most sophisticated case we can try to "expand" the surrounding correctly received frames by presenting them for some longer time (see Figure 11). This is certainly the best way as the viewer will not notice the discontinuity if many frames are just presented for a fraction of time longer than the regular frame. However, this method is not of practical value with the current video technology: There we always encounter well defined frame rates and we can not adjust just one window to have another frame rate than, e.g., 70 Hz.

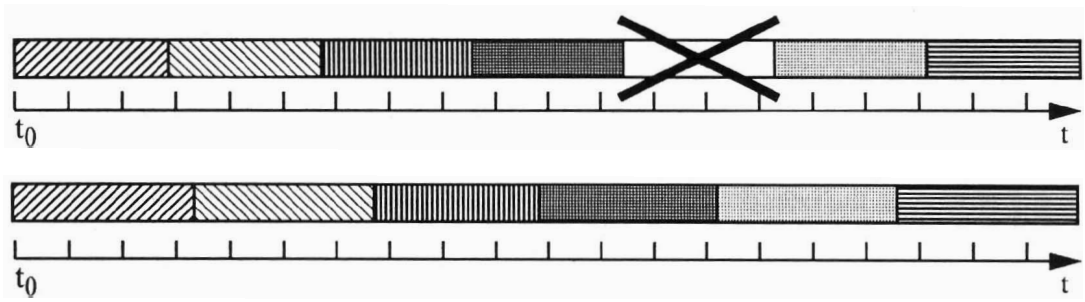

Figure 11: Expanding each frame

- As another technique one frame can just be doubled (see Figure 12). This is possible and the experiments showed that this is a good way to recover from lost video data.

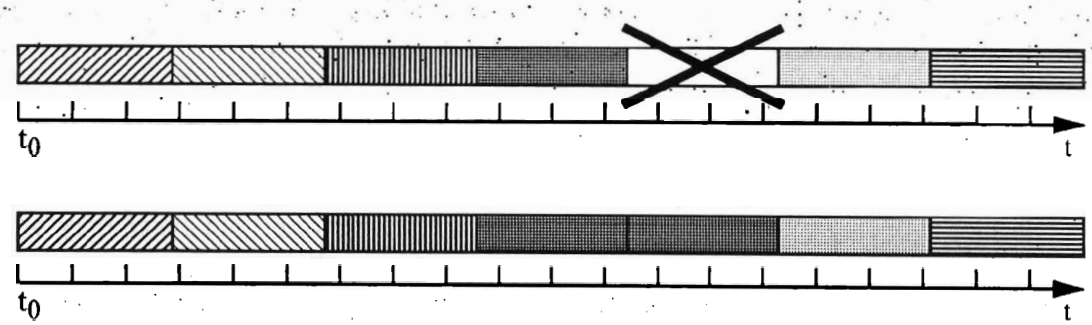

Figure 12: Continuing the data stream by doubling a frame

- The most common method is just to drop the corrupted frame and to continue with the next frame (see Figure 13). Initially we thought, that this is the worst solution. However, we did not found a significant difference by showing people the doubling technique shown in Fig-

ure 12 or this approach as shown in Figure 13.

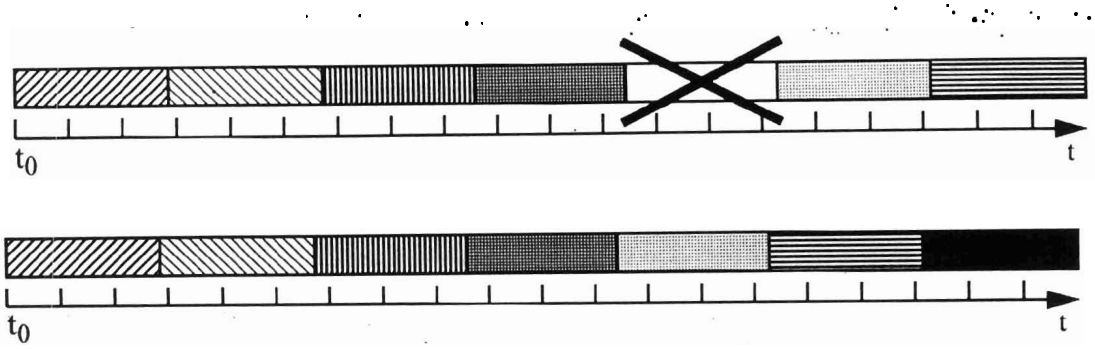

Figure 13: Continuing the data stream without doubling a frame

It turned out that the most important influencing factor is definitely the speed of objects in the image, i.e., whether there is a discussion with sitting participants in front of a static background or a car race going on.

Initially we thought, the faster the image is moving, the easier the recognizable jitter will be. That is not completely true: Only with very slow motion objects in front of a static background this is correct. Very slow means that, e.g., an object moves from left to right in 5 or more seconds. On the other hand jitter of domain can also be tolerated for scenes with very fast moving objects in front of static background, e.g., a tennis ball in a tennis game.

Hence jitter can be tolerated at the chance of scenes or if we encompass only a very fast or a very slow movement of the 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. This results can be used as: (1) The advances in video parsing makes us believe that we will soon be able to identify changes of scenes in real time [ZKSm93], and (2) we will also well be able to detect very slow and very fast motion in scenes in real time in the future.

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 most shorten 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 a continuous movement. For this second scenario initial experiments have shown that not more than 15 pointer updates per second are required.

As our envisaged **application of these results**, the knowledge of the skew (without any jitter) provides the means to adjust the buffers and control algorithm at the call set-up phase of multimedia data connections.

Also it is taken into account whenever an error on one of the two path occurs: Let us assume that so far there is no skew at the receiver. Then a packet at the 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 to be a continuous stream without any gap or skip to be done. As the consecutive 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, with the notion of where jitter will not be detected by the user, 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.

The notion of where we can tolerate 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. Then even with very accurate clocks after a certain time of, e.g., half an hour, we may encounter a difference of 33 ms which means that either the receiver buffer tends to cross the low water or a high water mark. At this point it would be nice to either introduce an artificial gap of one frame or to skip 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 we encounter a plethora of concurrent processes making use of the same scarce resources. A skew 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 a feather in our cap to reach results for wide range of media synchronization with extensive user interface experiments. The enforcement of which remains to be a different item which already has been addressed in several systems with dedicated synchronization system support or appropriate resource management components.

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. Clemens Engler 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 synchroniza-

tion related to music. 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, 1991.

[Blak92] Gerold Blakowski, Jens Huebel, Ulrike Langrehr, Max Muhlhaeuser: *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 26(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.

[DLW93] Bert J. Dempsey, Jorg Liebeherr, Alfred C. Weaver; A New Error Control Scheme for Packetized Voice over High-Speed Local Area Networks; Proceedings of 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.

[HeSt91] Ralf G. Herrtwich, Ralf.Steinmetz: *Towards Integrated Multimedia Systems: Why and How,* Informatik-Fachberichte, no. 293, Springer Verlag, 1991, pp. 327-342.

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

[LiGh90] 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.

[LiGh90] 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.

[LKGe92] L.Li, A. Karmouch, N.D. Georganas, *Synchonization in Real Time Multimedia Data Delivery*, IEEE ICC'92, Chicago, USA, June 1992. pp. 322.1.

[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

[Murp90] Alan Murphy: *Lip Synchronization Set of Experiments Conducted in Hursley, UK*, personal communication , 1990.

[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*, in 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 Synchronisation Required by Multimedia Applications*, Computer Communications, vol.13, no.7, September 1990, pp. 399-406.

[SRRa90] Ralf Steinmetz, Johannes Rueckert, Wilfried Racke: *Multimedia-Systeme*, Informatik Spektrum, Springer Verlag, vol. 13, no. 5, 1990, pp. 280-282.

[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*, IEEE Multimedia Workshop '92, Monterey, April 1992.

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

[Stei94] Ralf Steinmetz, Klara Nahrstedt; The Fundamentals in Multimedia, to appear at Prentice-Hall, inc., December 1994f

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

[StHe91] Ralf Steinmetz, Ralf Guido Herrtwich: *Integrated Distributed Multimedia-Systems* (in German), Informatik Spektrum, Springer Verlag, vol.14, no.5, October 1991, pp.280-282.

[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 acm SIGCOMM'91, Zuerich, Switzerland, September, 1991.

[ZKSm93] HongJiang Zhang, A. Kankanhalli, Stephen W. Smoliar; Automatic Parsing of Video; acm/Springer 'Multimedia Systems', 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'.

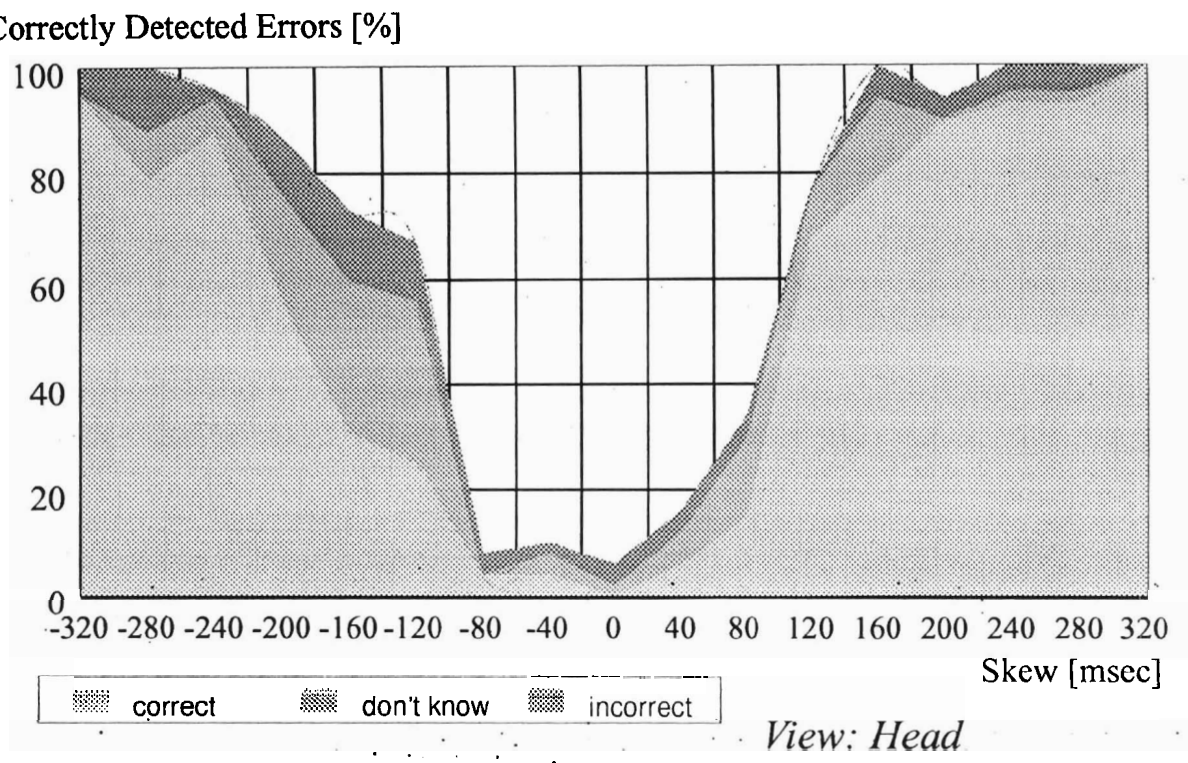

Figure 14: Correct detection of synchronization errors at head view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

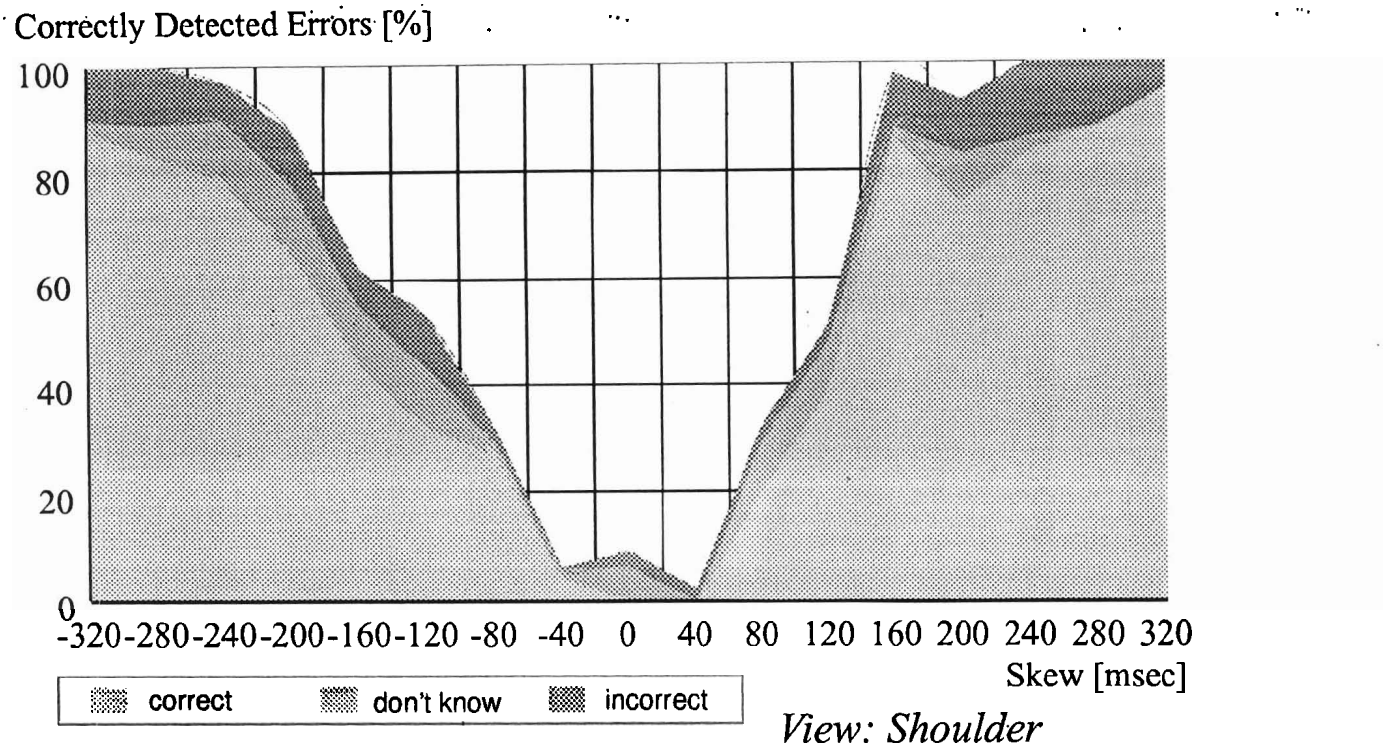

Figure 15: Correct detection of synchronization errors at shoulder view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

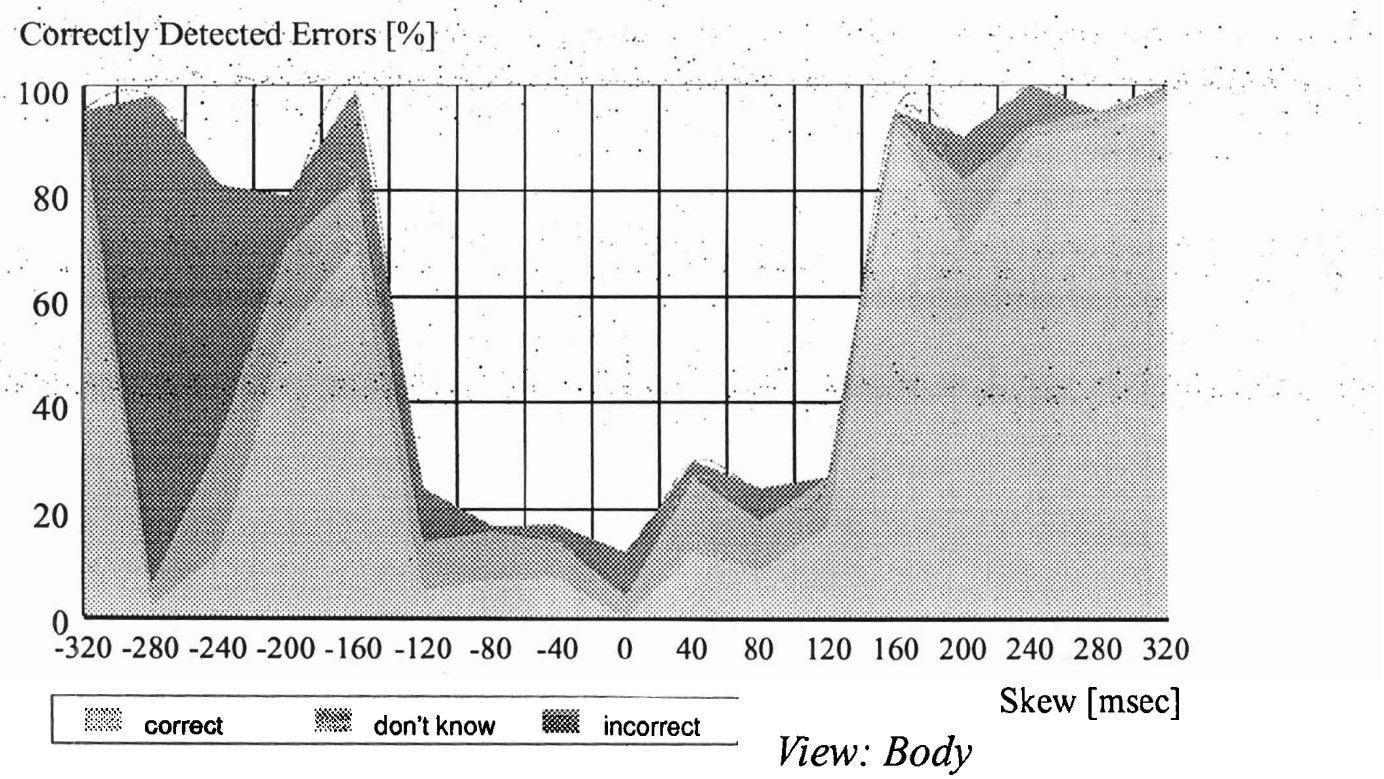

Figure 16: Correct detection of synchronization errors at body view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

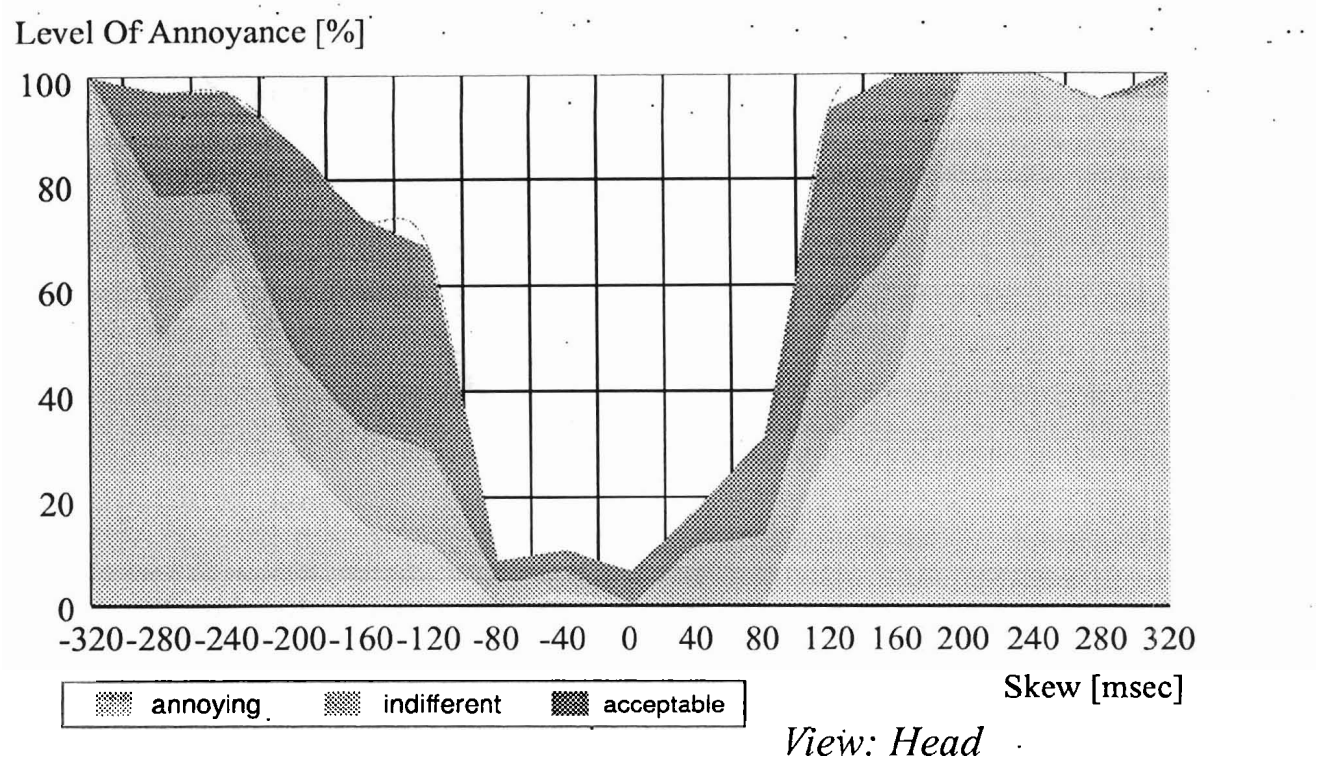

Figure 17: Level of Annoyance at head view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

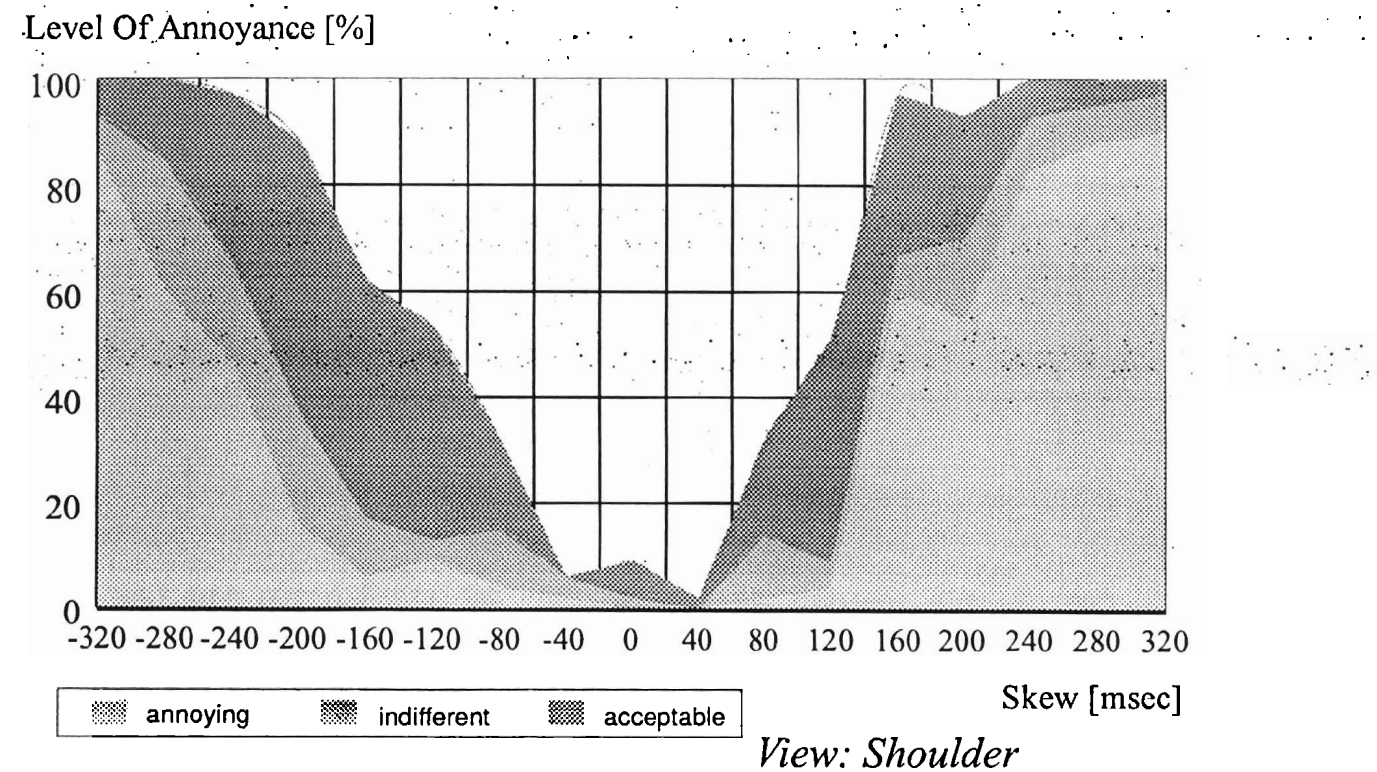

Figure 18: Level of Annoyance at shoulder view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

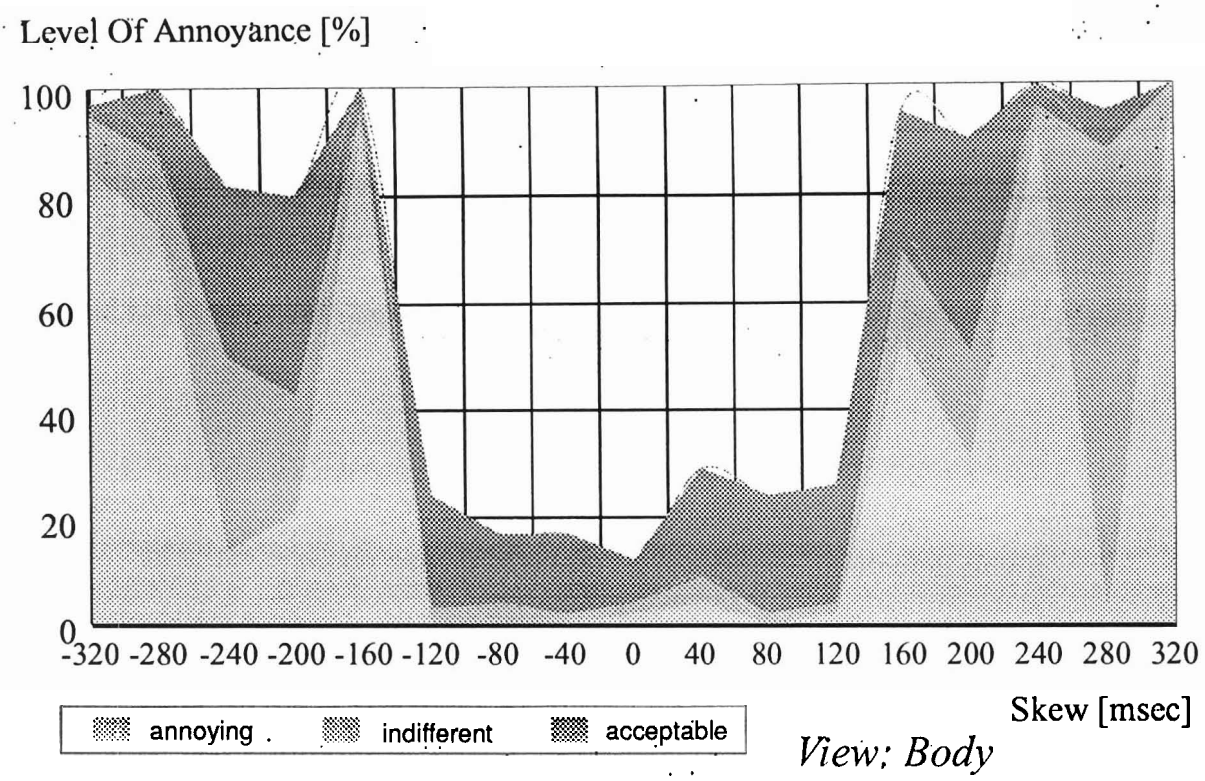

Figure 19: Level of Annoyance at body view
Left part, negative skew: video ahead of audio; right part, positive skew: video behind audio

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 synchronisation 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