

# Visualisation of Parallel Data Streams with Temporal Mosaics

Saturnino Luz

*Department of Computer Science  
Trinity College Dublin  
Dublin 2, Ireland  
luzs@cs.tcd.ie*

Masood Masoodian

*Department of Computer Science  
The University of Waikato  
Private Bag 3105, Hamilton, New Zealand  
m.masoodian@cs.waikato.ac.nz*

## Abstract

*Despite its popularity and widespread use, timeline visualisation suffers from shortcomings which limit its use for displaying multiple data streams when the number of streams increases to more than a handful. This paper presents the Temporal Mosaic technique for visualisation of parallel time-based streams which addresses some of these shortcomings. Temporal mosaics provide a compact way of representing parallel streams of events by allocating a fixed drawing area to time intervals and partitioning that area according to the number of concurrent events. A user study is presented which compares this technique to a standard timeline representation technique in which events are depicted as horizontal bars and multiple streams are drawn in parallel along a vertical axis. Results of this user study show that users of the temporal mosaic visualisation perform significantly better at detecting concurrency, interval overlaps and inactivity than users of standard timelines.*

## 1. Introduction

Visualisation of data against time in the form of time-series is perhaps the world's oldest, and most frequently used form of visualisation. As Tufte [19] points out

... the natural ordering of the time scale gives this design a strength and efficiency found in no other graphic arrangements.

Time-series graphs have been widely used since the late 1700's, and their use has ranged from visualisation of mathematical and statistical data to train timetables, travel itineraries [13, 12] and even movement of various types of animals [19]. In more recent years a variation of time-series, in the form of timelines, has become the basis for visualisation of multiple, parallel data streams mapped onto a time axis. Its intuitiveness and the fact that it facili-

tates individuation and comparison of events make timeline visualisations particularly useful for viewing multimedia data. Therefore, it is not surprising that media production applications such as Adobe Premiere™ [15], Macromedia Director™ [4] and others use timelines to support user interaction with time-based data. Other designs that use parallel-stream timelines include the well-known Gantt charts [7], which were originally developed for visualisation of project time management, time browsers [3] and *LifeLines* for visualisation of patient records [14], and interactive time bars for visualisation of audio [5] and recorded multimedia meeting contents [16, 11, 18].

One of the advantages of the time-series graphs, and therefore timelines, is their effectiveness in supporting visualisation of large amounts of data over long periods of time. Another strength of this type of graph is that they support viewing of the same data at different levels of detail. This makes timelines particularly attractive for interactive computer applications, since they allow the user to zoom into or out of the displayed data and are suitable for transformation by context+detail distortion techniques [6]. This feature of timeline visualisations is particularly important in the case of small devices such as mobile phones and PDAs, where screen real-estate is at a premium [11, 12, 1].

When conventional timeline graphs are used to visualise multiple data streams it is necessary to allocate parts of the graph (often in the form of parallel strips along the time axis) to individual data streams. In a timeline visualisation of the recorded multimedia meeting data, for instance, there are usually several data streams (e.g. audio, video, text) to be plotted for each of the meeting participants displayed in parallel with the time axis for the duration of the meeting [16, 10, 18]. This space requirement limits the usefulness of the standard timeline visualisation on small displays (e.g. on mobile devices) when the number of multiple data streams increases to more than a few.

In visual terms, the main tasks users perform when using a standard timeline representation of parallel streams are: to detect intervals containing overlapping events and

concurrency by lining up the horizontal bars along the vertical axis, to estimate duration by assessing the widths of those bars, and to detect inactivity intervals by identifying blank regions on the chart. These visual tasks can be regarded as variants of a basic alignment task which the human visual system happens to be particularly good at, exhibiting in fact what has been termed *visual hyperacuity* [22]. However, visual sensitivity to spacial alignment decreases as the distance between the targets increases [21]. This is specially the case of standard timeline representations of parallel media streams, where alignment often needs to be recognised across several parallel lines. This paper presents a visualisation design which supports visual hyperacuity by minimising the distance between the streams on the graph, depicting them as a *temporal mosaic*. Temporal mosaics have been successfully employed to display multimedia recorded meeting data with a greater number of participants than would be possible using the standard timeline, over the same amount of display space [11]. An empirical user evaluation comparing the two types of timeline visualisations clearly shows many advantages of the temporal mosaic method.

The paper is structured as follows. In the next section we introduce the main design concepts of Temporal Mosaic, motivating them through an example from visualisation of multimedia meeting records and contrasting them with those of standard timeline visualisation. In section 2.1 we describe an efficient algorithm for drawing temporal mosaics and illustrate its application. Section 3 describes an experiment designed to compare user performance on a range of visual tasks when using temporal mosaics and standard timelines, and presents the results and statistical analysis of the experiment. The paper closes with a discussion of the prospects of temporal mosaics as a visualisation technique, suggestions of potential applications, and general conclusions.

## 2. The Temporal Mosaic

As we have seen, most interfaces for visualisation of parallel streams of time-based activities, such as speech and video sequences, adopt a timeline metaphor in which events are depicted as horizontal (or, less commonly, vertical) bars stretching along a temporal axis. We henceforth refer to this visualisation style as *standard timeline visualisation*. A typical example of standard timeline visualisation is the depiction of speech, text editing, and other events in meeting browsers [20, 16]. Figure 1 illustrates this technique by representing a period of communication between four users over two media: audio and text.

It can be readily verified in Figure 1 that standard timelines tend to require a considerable amount of space since inactivity intervals for each stream need to be represented as

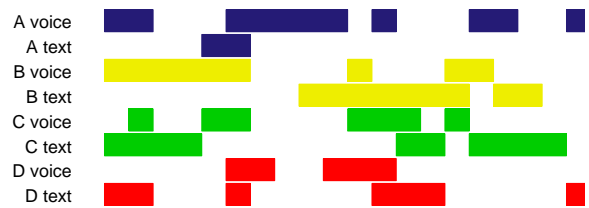


Figure 1. Standard timeline visualisation

blank areas. The graph grows vertically with the number of streams, potentially impairing the user's ability to recognise concurrent events (across streams) and exclusively inactive intervals. For certain tasks, such as browsing of meeting recordings, being able to recognise and locate these kinds of events can be crucial [2]. The temporal mosaic was initially developed to address these issues.

Temporal mosaics allocate space proportionally so as to show the contributions of each participant to each stream interval. Thus a single horizontal bar is required per stream. Figure 2 depicts the same interval as shown in Figure 1 as a temporal mosaic. The top bar represents speech. Rectangles that stretch through the entire height of the bar represent speech from a single source (speaker) and are painted with the colour corresponding to that source. When the floor is shared between two or more speakers during a given interval, the rectangle for that interval is split vertically into rectangles of height inversely proportional to the number of contributing speakers. The mosaic-style visualisation therefore makes better use of screen space by providing a more compact, and arguably more natural representation of silences and text inactivity while preserving the linearity constraints typical of standard timelines.

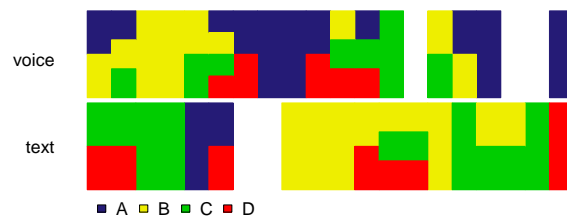


Figure 2. Temporal mosaic visualisation

### 2.1. Temporal Mosaic Drawing Algorithm

In order to draw a standard timeline or a temporal mosaic we assume that activity intervals have been discretised and encoded as an  $n \times m$  activity matrix  $M$  where each column represents an activity stream. These can be viewed as

binary-valued vectors where each element corresponds to a tick of the clock: 1's represent activity and 0's represent inactivity for the atomic interval delimited by a particular tick and the preceding one. The algorithm for drawing standard timelines is straightforward. One needs simply scan each column of  $M$  marking pairs of positions which delimit contiguous activity intervals. Each of these pairs represent the  $x$  coordinates (left and right corners) of a rectangle to be plotted. The  $y$  coordinates (top and bottom corners) are individually assigned from a sequence  $(1, 2, \dots, m, m + 1)$  to columns of  $M$ . Rectangles corresponding to the first column would thus have as (bottom-left and top-right) coordinates 4-tuples of the form  $(x_l, 1, x_r, 2)$ , rectangles for the second column would be plotted at  $(x_l, 3, x_r, 4)$  and so on.

Plotting the temporal mosaic is a somewhat more involved process than drawing standard multi-stream timelines. In the mosaic visualisation, each atomic interval could, in principle, be coloured in  $2^n$  distinct ways, where  $n$  is the number of streams to be represented. Since an interval of  $n$  concurrent activities will be depicted in the mosaic as  $n$  rectangles of equal height, for each interval one needs to know in advance of drawing which streams will be depicted as active. A simple approach to drawing the mosaic could be to test each atomic interval for concurrency and plot as many rectangles of width one as there are active streams. Thus a 12-tick interval consisting of 4 concurrent streams, for instance, would require 48 separate rectangle drawing operations. Multiple drawing operations are however a source of inefficiency which could make plotting of mosaic visualisations impractical for large, fine-grained intervals.

Our implementation employs a more efficient method which is described in Algorithm 1.

#### Algorithm 1 Mosaic drawing.

```

1 function drawMosaic(M, P)
2   input: M      /* an n x m activity matrix */
3         P      /* the first m primes */
4   var: R ← (1, ..., 1) /* a vector of size n */
5        X      /* a pair of x coordinates */
6        T      /* types of mosaic rectangles */
7   for (i in 1:m)
8     for (j in 1:n)
9       Mi,j ← Mi,j × Pi
10      if (Mi,j = 0)
11        Mi,j ← 1
12      Rj ← Rj × Mi,j
13   T ← unique(R) / {1}
14   for each (t in T)
15     X ← getHorizontalCoordinates(R, t)
16     f ← factorise(t) /* factors of t */
17     v ← 1/length(f) /* height of sections */
18     for (i in 1:length(f))
19       setColour(fi)
20     drawRectangle(X1,  $\frac{i-1}{v}$ , X2,  $\frac{i}{v}$ )

```

This method basically consists of assigning 1's to inactive intervals (line 11) and identifying active segments of

each stream vector by a (unique) prime number (line 9), and subsequently merging all streams into a single vector (R) by multiplying all individual vectors (line 12). The resulting vector will contain 1's representing overall inactivity and multiples of the initial prime numbers standing for different combinations of concurrently active streams. Figure 3 depicts this process: the top three streams are represented as primes and compactly encoded as their multiples (1, 5, 30, etc) in the stream shown at the bottom, which corresponds to the temporal mosaic representation of these three primary streams. The next step is to retrieve the coordinates for the various rectangles representing the different combinations, which can be done by stepping through the vector of multiples, where the length of each segment of integers greater than 1 corresponds to the width of a rectangular area shared by rectangles whose heights and colours will be determined by the factors of the integer assigned to the segment (lines 16 and 17). Finally, the individually-coloured rectangles are drawn at the appropriate heights (lines 18–20).

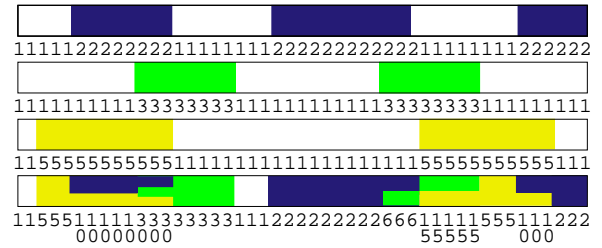


Figure 3. Activity streams encoded as primes and multiples

### 3. Comparing Mosaic and Timeline

A user study was conducted to compare the mosaic and timeline methods in visualisation tasks involving information related to recorded multimedia meeting contents. Although this study focused on visualisation of recorded meeting contents, its results are applicable to other uses of the timeline and mosaic visualisations.

#### 3.1. Methodology

The study was conducted over the Internet and employed a between-subjects design. The subjects accessed a web page which randomly assigned them to one of the two visualisations (i.e. standard timeline, or temporal mosaic). The subjects were then given a written tutorial on the visualisation to which they had been assigned. This tutorial consisted of a short text which described the form of visualisation to be used, illustrating it with graphs similar to

those actually used in the experiment. At the end of this tutorial the subjects were directed to the appropriate visualisation trial. The visualisation test was implemented using a Java WebStart™ application, which presented 26 questions randomly and one at a time to the subjects<sup>1</sup>. For each question once the subject had selected their response from a list of choices, the system recorded the time taken to answer the question, and then presented the subject with a ranking scale of 1 to 7 (1 being easy and 7 being difficult) requesting their rating of the difficulty of the question they had just answered.

The questionnaire used in this study comprised of questions about visualisations containing 4, 8 and 12 streams from recorded collaborations over speech and text [9] by groups of 2, 4 and 6 participants, respectively). Of these, 8 questions referred to the 4-stream visualisation, while the 8- and 12-stream visualisations received 9 questions each. The questions assessed how well the subjects were able to detect the following events<sup>2</sup>:

- (e) exclusion: detection of activities performed by a single participant,
- (s) single-media concurrence: detection of concurrent activities by different participants on a single medium,
- (m) multiple-media concurrence: detection of concurrent activities by the same participant on different media,
- (i) inactivity: detection of silence or text inactivity, and
- (d) duration: assessment of duration of sets of events

Table 1 shows examples of the questions used for the 4-stream condition, illustrating each of the above described types.

Thirty subjects took part in this study, half of whom were assigned to each of the two experimental conditions. Ten of the subjects were females and 20 of them males; with 15 being students, 12 academics and 3 others. The age of the subjects ranged from mid-twenties to mid-fifties (15 in 20-29, 7 in 30-39, 4 in 40-49, 4 in 50-59).

### 3.2. Results

The evaluation compared the mosaic and the standard timeline visualisations with respect to accuracy, answer time and perceived difficulty. Accuracy was calculated as the ratio of the number of correct answers to the total number of questions asked per subject. Answer time comparisons considered the average time it took each subject to

<sup>1</sup>The two versions of the tutorial and experiment can be accessed on-line at <http://ronaldo.cs.tcd.ie/~luzs/timeline/> and <http://ronaldo.cs.tcd.ie/~luzs/mosaic>.

<sup>2</sup>Abbreviations shown in brackets will be used as labels for the different events on the graphs below.

**Table 1. Sample questions**

Question	Type
How many times (if any) did B speak on his own?	e
How many times (if any) was there complete silence (i.e. no speaking by anyone)?	i
How many times (if any) did A speak at the same time as D (and possibly others)?	s
Who spent more time editing text: A or D?	d
How many times (if any) did A, B, C and D speak at the same time?	s
How many times (if any) did B edit text at the same time as D (and possibly others)?	s
Who spoke more frequently: A or D?	d
How many times (if any) was there no editing of text by anyone?	i
How many times did B speak and edit text at the same time?	m

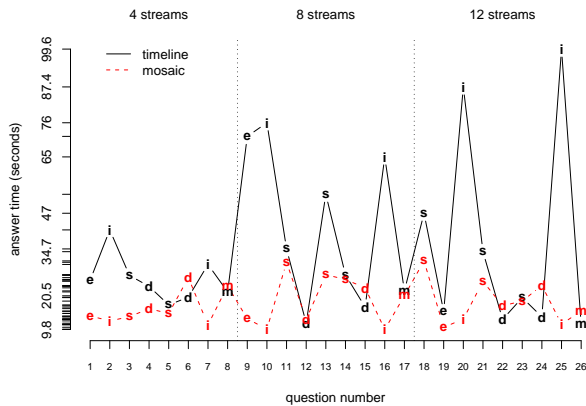
answer each question. Perceived difficulty was measured in terms of the subjects' rating of questions on a Likert scale ranging from 1 to 7, as described above.

**Table 2. Summary results of comparisons.**

	mosaic	timeline	signif.
Accuracy (% correct answers)	84.10	65.89	**
Time (seconds)	18.24	35.74	**
Rating (1-7 scale)	3.19	3.34	not

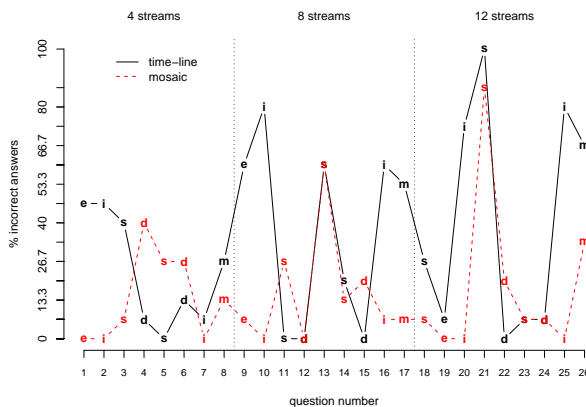
Mosaic users performed significantly better than standard timeline users both in terms of accuracy and answer time. However, no difference was found with respect to difficulty ratings. The overall results are summarised in Table 2. On average, mosaic users answered questions nearly twice as fast than standard timeline users (Welch Two Sample t-test indicated  $p < 0.0001$  with a 99% confidence interval of 12.7 to 22.3) while getting over 84.1% of the answers right, against a timeline accuracy of 65.9% ( $p < 0.0001$ , 99% confidence interval: 10 to 26). Although the difficulty ratings for mosaic were marginally lower than for timeline, this difference was not found to be statistically significant. This is probably due to the fact that, not having seen an alternative visualisation style, individual subjects tended to rate difficulty against other questions within the same visualisation style. This hypothesis is supported by the fact that we found strong correlation between mean ratings and answer times for both mosaic and timeline (Person's product-moment correlation of 0.74 and 0.95, respectively, with  $p < 0.001$ ). It is interesting to note that the correlation is weaker for mosaic, possibly reflecting the fact

that since mosaic users were not familiar with the mosaic-style visualisation at the beginning they rated the questions less consistently than timeline users.



**Figure 4. Answer times for mosaic and timeline visualisations**

Detailed breakdowns of answer times and percentage of errors for each condition evaluated are given in Figures 4 and 5. Each of these charts is divided into three sections delimited by dotted vertical lines. These sections represent the densities of the visualisations presented to the subjects with respect to the number of streams depicted, and cover the full range of question types described in section 3.1. Question types are identified in the chart by individual letters (*e* stands for exclusion, *i* for inactivity, etc).



**Figure 5. Error rates for mosaic and timeline visualisations**

The segmentation by number of streams highlights the fact that performance in general degrades as the number of

streams increases, though that effect is less acute for subjects using the mosaic visualisations. In fact, answer time hardly varied across the three conditions for these subjects, which attests to the scalability of the temporal mosaic design. Although mosaic outperformed timeline for nearly all question types, the greatest differences were found in detection of inactivity (*i*), exclusion (*e*) and single-media concurrence (*s*).

Error rates (Figure 5) correlate with answer time. Again, the greatest gains for temporal mosaic users in terms of accuracy can be found in inactivity, exclusion and, to a lesser extent, single-media concurrency detection. In addition, for multiple-media concurrency (*m*) detection, despite the fact that subjects using the temporal mosaic did not outperform timeline users in terms of answer times, the former were considerably more accurate (17.8% versus 49% error rate,  $p < 0.002$ ). The only type of question for which timeline users outperformed temporal mosaic users (though by a small margin, not found to be statistically significant) were questions involving assessment of event duration (*d*) in the 4-stream condition. This is very likely due to some subjects mistakenly assuming direct proportionality between rectangle area and event duration in the mosaic visualisation, when the relation that holds is in fact between event duration and rectangle width.

Another fact worth noting is the discrepancy between the short time it took subjects using the mosaic visualisation to answer questions about single-media concurrence (*s*) and the relatively high error rates recorded for these responses. Closer inspection of the answer logs suggests that the inaccuracies were caused by failure of some subjects in individuating events in the way intended. If two users wrote at the same time for an uninterrupted interval, that should have counted as a single instance of concurrence event, regardless of whether other users also wrote during a sub-interval. However, in cases where a shorter intervention by other users took place, subjects often chose to count the interval as two distinct events. If that is in fact the case, then answer time would be a better indicator of effectiveness for single-media concurrence tasks.

### 3.3. Discussion

The user evaluation has clearly demonstrated the advantages of the temporal mosaic over the standard timeline visualisation, particularly when the number of parallel streams of data increases to more than 4. The results are even more encouraging considering the fact that most of the study participants are likely to have been more familiar with timeline visualisation than the with novel temporal mosaic visualisation.

In cases where screen real-estate is limited (e.g. on mobile phones and PDAs) temporal mosaic visualisation

would be more effective as it requires less space than the timeline visualisation for the same number of data streams.

There are also several ways of improving the timeline visualisation itself, if it is preferred over temporal mosaic due to reasons such as the users' familiarity with it. For instance, associating colours to individual media rather than each of the participants in recorded meetings visualisation would make it easier for the user to scan over activities related to a particular medium (e.g. text inactivity, or concurrent speech by two participants). Such variants are an interesting object for further research.

## 4. Conclusions

This paper presented a novel technique for visualisation of parallel time-based data streams based on the concept of temporal mosaic. Although mosaic-style visualisations have been used before for visualisation of cross-tabulated data [8] and exploratory data analysis [17], the temporal mosaic technique proposed in this paper is the first to explore partitioning of the display area to represent concurrency on a time axis. Evaluation showed that temporal mosaic users outperform standard timeline users in most concurrency detection tasks, thus confirming our hypothesis that the more compact mosaic design would favour visual hyperacuity. Although this paper illustrated the technique through an application to recorded meetings, we believe temporal mosaics will find natural applications in a variety of situations involving the display of concurrent events and similar time-based inter-dependencies.

## References

- [1] B. B. Bederson, A. Clamage, M. P. Czerwinski, and G. G. Robertson. DateLens: A fisheye calendar interface for PDAs. *ACM Transactions on Computer-Human Interaction*, 11(1):90–119, 2004.
- [2] M.-M. Bouamrane, S. Luz, and M. Masoodian. History based visual mining of semi-structured audio and text. In *Proceedings of the 12th International Multi-media Modelling Conference, MMM06*, pages 360–363, Beijing, China, Jan. 2006. IEEE Press.
- [3] S. Cousins and M. Kahn. The visual display of temporal information. *Artificial Intelligence in Medicine*, 3(6):341–357, 1991.
- [4] Macromedia director. <http://www.adobe.com/products/director/>, March 2007.
- [5] J. Foote. An overview of audio information retrieval. *Multi-media Systems*, 7(1):2–10, 1999.
- [6] G. W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, New York, NY, USA, 1986. ACM Press.
- [7] H. Gantt. Organizing for work. *Industrial Management*, 58:89–93, 1919.
- [8] J. Hartigan and B. Kleiner. Mosaics for contingency tables. In *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, pages 268–273, 1981.
- [9] S. Luz, M.-M. Bouamrane, and M. Masoodian. Gathering a corpus of multimodal computer-mediated meetings with focus on text and audio interaction. In N. Calzolari, editor, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2006*, pages 407–412, Genoa, Italy, 2006.
- [10] S. Luz and M. Masoodian. Compact visualisation of multimedia interaction records. In *Proceedings of the 7th International Conference on Information Visualisation (IV '03)*, pages 536–541, London, UK, 2003. IEEE Computer Society.
- [11] S. Luz and M. Masoodian. A mobile system for non-linear access to time-based data. In *Proceedings of Advanced Visual Interfaces AVI'04*, pages 454–457. ACM Press, 2004.
- [12] M. Masoodian and D. Budd. Visualization of travel itinerary information on PDAs. In *AUIC'04: Proceedings of the fifth conference on Australasian user interface*, pages 65–71, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
- [13] M. Masoodian and N. Lane. An empirical study of textual and graphical travel itinerary visualization using mobile phones. In *AUIC'03: Proceedings of the Fourth Australasian user interface conference on User interfaces 2003*, pages 11–18, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.
- [14] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: visualizing personal histories. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 221–227, New York, NY, USA, 1996. ACM Press.
- [15] Adobe premiere. <http://www.adobe.com/products/premiere/>, March 2007.
- [16] D. M. Roy and S. Luz. Audio meeting history tool: Interactive graphical user-support for virtual audio meetings. In *Proceedings of the ESCA workshop: Accessing information in spoken audio*, pages 107–110. Cambridge University, Apr. 1999.
- [17] R. Spence. *Information visualization*. Addison-Wesley Harlow, 2001.
- [18] S. Tucker and S. Whittaker. Accessing multimodal meeting data: Systems, problems and possibilities. In *MLMI '04: Machine Learning for Multimodal Interaction*, pages 1–11. Springer-Verlag GmbH, 2005.
- [19] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2nd edition, 2001.
- [20] A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen. Meeting browser: Tracking and summarizing meetings. In D. E. M. Penrose, editor, *Proceedings of the Broadcast News Transcription and Understanding Workshop*, pages 281–286, Lansdowne, Virginia, Feb. 1998. Morgan Kaufmann.
- [21] S. J. Waugh and D. M. Levi. Spatial alignment across gaps: contributions of orientation and spatial scale. *J Opt Soc Am A Opt Image Sci Vis*, 12(10):2305–2317, Oct 1995.
- [22] G. Westheimer and S. P. McKee. Spatial configurations for visual hyperacuity. *Vision Res.*, 17(8):941–947, 1977.