Auditory Direct Manipulation of Acoustical Objects by Blind Computer Users

Ludger Bölke and Peter Gorny
University of Oldenburg, Informatics Department
Computer Graphics & Software Ergonomics Unit
Postbox 2503, D-26111 Oldenburg, Germany
Phone: +49 441 798-4516 or -2901 - Fax: +49 441 798-2155
EMail: Ludger.boelke@.informatik.uni-oldenburg.de
Gorny@.informatik.uni-oldenburg.de

Abstract

Several recently developed systems allow blind users to work with auditory representations of graphical user interfaces (GUI). But these adaptations do not provide sufficient feedback for the blind enabling them to handle the mouse in an effective way. In this paper the direct manipulation of acoustical objects ,"hearcons", is described. An acoustical feedback is provided to control the mouse movements and to select the target object. The basis for this ear-hand-coordination is the combination of permanently active hearcons and acoustical feedback indicating the relative position of the mouse cursor with regard to the objects. By making use of spatialized sound the hearcons can be placed anywhere in the space and the blind user can arrange the objects to his own needs. Beyond the adaptational approach we describe an assistive interface for which our methods will be implemented in a window system to utilize the advantages of the GUIs for blind users .

Keywords acoustical feedback, hearcons, direct manipulation, blind users, ear-hand-coordination, auditory interface

1. Introduction

Blind persons have to compensate their visual impairment by tactile and acoustical perception. Therefore speech synthesis (dominating in the U.S.) and Braille-output (dominating in Germany) have been developed to enable blind people to use the computer. These adaptive technologies base on screenreaders, which extract the outputs of the application programs directly from memory. In text-based systems the characters shown on the screen are stored as ASCII-signs in a text-buffer, which can be transformed directly into Braille-signs or into speech. In graphics-based computers the screen output is stored as a bitmap in memory (excluding gray-scale and color coding). A bit contains the information whether a pixel on the screen must be set or not. When viewing the pixels on the screen a user perceives them as text or graphics. The graphics based systems form the technical basis for the graphical user interfaces which allow a more intuitive use. Objects and functions are represented in textual or pictorial form directly on the screen and can be referenced by selecting them with a mouse click, and by moving them with the mouse (drag and drop).

In order to retrieve the textual information from the bitmap special pattern recognition programs are needed, which are time and resource consuming and are still not able to recognize every font. Therefore blind users were excluded from using graphical user interfaces for a long time.

A solution to this problem is the off-screen model [Schwerdtfeger91]. The off-screen model collects every screen output in ASCII-characters. To build this model every function concerning the screen output is intercepted and the information about each object is stored in a database. In the textual information about the currently touched object is sent to the speech synthesizer or the Braille-line.

The following chapter describes in detail the three problems blind people are confronted with when working with graphical user interfaces. We will show, that a blind cannot work with existing adaptations of the graphical user interfaces as effectively as sighted users do.

We will then explain the direct manipulation of acoustical objects. This technique enables blind users to apply the mouse for an effective direct manipulation of screen objects. It is the basis for our assistive system SPUI-B (StereoPhonic User Interface for the Blind).

2. Graphical user interfaces and blind users

In this chapter we will investigate the advantages of the graphical user interfaces (GUI), which makes them so appealing for sighted users, before reporting the problems blind persons are confronted with when using them.

2.1 GUIs and their barriers for blind users

The advantages of GUIs (versus textual interfaces with command and menu-oriented control) have been discussed widely during the last decade. We refrain from repeating all arguments and reasons and list only the four most important benefits in the context of our subject:

1. The use of metaphores related to the customary user environment visually represented by icons, which are references to data structures, programs, functions, etc.;
2. Direct manipulation of these visible objects (icons) with rapid visible feedback for all actions;
3. Ease of control of simultaneously available programs and functions;
4. Structuring and organizing visible objects topographically on the screen area by linking semantics to their positions.

GUIs aim at enhancing the usability of computer systems for non-computer-specialized users for their normal work tasks, but they fail completely for visually impaired users - on the contrary: they cause new hindrances for blind users of computer systems.

Boyd et al. [Boyd91] stated the following three problems - called barriers - which cause difficulties for blind users to work with GUIs in the same way as sighted users do:

1. The *pixel barrier* refers to the problem that the screen output is stored as a pixel map in memory and cannot be read by usual screenreaders. A solution for this technical problem is the off-screen model mentioned above.
2. The *mouse barrier* refers to the problem that blind users cannot handle the mouse in an effective way as an input device; this is a motorial problem resulting from the lack of feedback. Sighted people are able to use the mouse because they use the *eye-hand-coordination* to control mouse movements. They *see* in which way the relative position of the cursor to the objects changes when moving the mouse in a certain direction. Blind people get no feedback about the position of the mouse cursor and therefore they cannot work with the GUIs in the same way as the sighted do. In the next chapter we will present an acoustical for this.
3. The *graphics barrier* is the most difficult problem and is located on the semantical level. It refers to the fact, that information is presented in a graphical manner, and that topography and topology give additional hints about the represented objects and their relations. The transformation of these graphical representations will always be accompanied with a loss of

tactile display, but it has the disadvantage, that such a devices has only a small resolution (approx. 8 dots per inch or 3 dots per mm) and is very expensive.

As already stated the problem of the graphics barrier is a semantical one and it is questionable whether it will ever be sufficiently solved. Graphics are used to represent an object in an obvious way. For example many icons are simplified pictures of objects the sighted users *see* in their everyday life such as the trashcan or the folder. If such icons are transformed into an tactile output - with a smaller resolution than the screen - it is doubtful whether blind users are able to interpret the icon with the same ease, because they know the objects only from touching them in their original size. The diminuished tactile form may be not only ambiguous, but it looses completely the metaphorical link to the real environment.

Blind users would have to learn the meaning of tactile icons as abstract representations and this learning process contradicts the goal of GUIs to be intuitively and obviously applicable. This disadvantage cannot be overcome by the development of cheaper two-dimensional tactile displays with greater resolution than the presently available displays [Fricke92], though these displays will certainly enhance the possibilities for blind users to work interactively with schematic representations of objects mapped in line drawings.

Some projects exist which try to transform graphics into acoustical output [Smith90] or to describe pictures for the blind [GUIB93], but there still exists no practical solution to represent any graphics via an acoustical medium.

2.2 Adaptations of given GUIs for blind users

The present adaptations do not solve these three barrier problems. Generally they crack the pixel barrier by using the off-screen model and by mapping visual objects onto sound objects positioned in an auditory three-dimensional space. Often blind users only have the possibility to use the cursor keys to navigate through the GUI.

For example the GUIB project (Textual and Graphical User Interfaces for Blind People) [GUIB93] transforms the screen contents of *MS-Windows* on the one hand into a tactile form, keeping the topographical arrangement of the objects. On the other hand the tactile output is supported by acoustical output with spatialized sound. A sound is attached to every object of the GUI and also to the cursor. The sound sources seem to be located in the same position as the visual objects on the screen. So, if the cursor moves, its sound source seems to move in the same direction too. Thus the user gets the information about the absolut cursor position on the screen. The other objects only sound when they are touched by the cursor. No direct selection of an object is possible, because no feedback is given about the relative position of the cursor to a target object when moving the mouse. This situation is comparable with a black screen where only the cursor is visible and a sighted user has to find a certain object on the screen which will light up when touched - an unsatisfying situation.

The SonicFinder [Gaver89, Gaver90, Gaver91], an adaption to make the Finder interface of the Macintosh accessible to visually impaired users, acts in the same way. The objects are implemented as acoustically passive objects which only sound when a certain event has happened. In a later version of the SonicFinder *soundholders* were introduced, which can be attached to an object and which sound permanently to provide a cue of its relative position that is acoustically represented by the volume reciprocally to the distance from the cursor: If the cursor moves away from the object the sound is turned lower. No spatialized sound is used, so that the soundholder cannot be located in order to move the mouse in a certain direction. The original problem remains: how to find an object before a soundholder can be attached to it.

The Soundtrack project [Edwards 88, Edwards89] implemented the absolute locations of objects by different tones (organ-keyboard-metaphore), but as tests proved, only a blind musician was able to retrieve the positional cue from the tonal coding, the other visually impaired test persons counted the objects when passing by with the cursor.

In the Mercator project [Mynatt92] *XWindows* under *UNIX* is adapted. The authors solve the navigation problem by replacing the graphical display with a tree-like hierarchical auditory interface. To select a certain object the tree has to be searched by using the cursor keys until the target object is found ("down" to move to a lower hierarchy, "right" to select a branch, "up" and "left" for the reverse operations). The objects again only sound when a certain event has happened.

As a conclusion we may state that all known adaptations of GUIs for blind users do not solve all three problems mentioned above. They do not enable blind users to apply the GUIs in the same way as the sighted do. Indeed only the pixel barrier has completely been overcome by using the off-screen model. In practice the GUI is reduced to a textual menu/command-oriented interface using cursor keys for navigation. The topographical arrangement of the objects is still of no meaning for the blind users. Direct manipulation of objects is not possible, since no sufficient feedback is provided to use the mouse adequately. This feedback must be perceivable in a direct manner and not indirectly by distinguishing different locations, for example with the organ-keyboard-metaphore.

3. Direct manipulation of acoustical objects

The basis for auditory direct manipulation are the acoustically active objects, called *hearcons*, positioned in a three-dimensional space, and the acoustical feedback for the *ear-hand-coordination* to control the cursor movements via a 3-D-mouse (or ather suited 3D-pointing device).

3.1 Hearcons

The idea is to represent an object of an interface by a sound, which is called a *hearcon* . We do not reduce the number of sounds, as it is done by Gaver [Gaver91], which implements environmental sound for the *auditory icons* in the *SonicFinder* , or like Blattner [Blattner89], who uses temperated tones for their *earcons* . We propose to use any sound which might support a certain application.

The difference to given acoustical adaptations of GUIs is, that the hearcons are realized as acoustically active objects which permanently sound and not only in case of an event. These acoustically active hearcons form the basis for the acoustical feedback when using the mouse. By making use of spatialized sound, realized by headrelated stereophony, the user can place a hearcon anywhere in the space. When all hearcons sound, the topographical arrangement of the objects can be heard. A hearcon is characterized by:

- the sound or the tones, which represent the object,
- the volume,
- the position coordinates in space,
- the size.

The coordinates of the hearcon give the position in space. The size of the hearcon is an artificial size is needed to bound the spatial volume of the hearcon when it is considered to be selected by the mouse cursor.

One hearcon is distinguished to be the "actual hearcon". Only the contents or outputs of the object represented by the actual hearcon can be perceived by the user, because only these informations are sent to the speech synthesizer[1] or the Braille-line. All other hearcons represent their objects only acoustically. This is comparable to a situation in a GUI, where all but the actual window is represented by an icon and only the output of the actual window is shown on the screen. this solution avoids also the "unselected window problem" of GUIs, because the user is never in doubt, which window (represented by the hearcon) an input will go to.

3.2 To select a hearcon

The approach by acoustically active hearcons offer the possibility to select a target hearcon directly without searching the whole space. A grouping of objects will allow to turn off all presently irrelevant groups of hearcons and enhance the possible selection process (implementing the GUI-feature "send to background"). The user *hears* the topographical arrangement of the objects and notices, in which direction the mouse must be moved to reach the target object.

To solve the mouse barrier, the ear-hand-coordination, which controls the hand movements to position the mouse cursor, must be implemented. The user needs a permanent and immediate acoustical feedback, in which way the relative position of the cursor to the target object changes when moving the mouse. Therefore the acoustical output of every hearcon changes when the mouse cursor is moved.

The following solutions for this acoustical feedback are investigated:

1. Every change of the cursor position is announced by changing the acoustical parameters of the hearcons such as pitch, loudness or timbre. For example a hearcon sounds clearer or louder when the cursor moves towards it and it sounds lower or muffled when the cursor moves away.
2. A movement of the cursor results in a change of the position of the hearcons in space. The acoustical room seems to move beneath the cursor, which has a fixed center position (just as the situation in a vehicle simulator, where the vehicle nose is always pointing to the center of the screen)[2].
3. A sound is attached to the cursor and this sound source seems to move according to the cursor movements. The output of the hearcons sound constantly without variation. The user hears the cursor moving towards another hearcon.
4. A movement of the cursor results in changes of the acoustical parameters of all hearcons and the sound source of the cursor seems to move, too. This is a combination of 1 and 3.

Which solution is best suited to control the ear-hand-coordination and which supports the selection of a hearcon in the best way for a given application, has to be further investigated in a series of experiments, which are performed in close collaboration with colleagues in the departments of Physics and Psychology of our university.

The touching of a hearcon must be announced immediately. The *AudioWindows* system [Cohen91, Cohen93] uses a "spotlight", when a window is selected. The idea behind the spotlight is the 'just noticeable difference', an important topic of the psychophysics: it is the smallest

[1]While listening to the output of the actual object, the other hearcons must sound lower, so that the user's attention is not turned away from the actual outputs. On the other side the hearcons must sound with their usual volume, when the user for example wants

noticeable perceptional (acoustical, visible, tactile) distinction between two objects [Roederer77]. Beneath this threshold only one object or two identical objects are perceived. Another solution is an additional tone which sounds, when the hearcon boundaries are crossed.

The same problems occurs when a selected hearcon must be marked. Either the acoustical output changes in a just noticeable way or an additional tone is introduced to announce, that a hearcon is marked.

3.3 To manipulate hearcons

In the following we describe shortly some more functions to handle hearcons.

Moving a hearcon

The moving of a hearcon takes place just the same way as in GUIs. A hearcon must be selected, marked and can be dragged to the target position. While dragging the hearcon, an immediate acoustical feedback must be given where in space the hearcon currently is. For that reason the sound source of the dragged hearcon will move according to the position of the cursor.

Creating a hearcon

Before a hearcon can be placed anywhere in the space, it must be created. For that purpose the user has to evoke the function 'create hearcon' by a doubleclick with the mouse button, when the cursor is not placed on any hearcon, or by striking a hotkey. The position of the new hearcon can be

- a fixed position for every new hearcon, for example, directly in front of the user, or
- the position of the mouse cursor when starting the creation process.

After the new hearcon has been created the cursor is implicitely placed on this hearcon, in order to enable the user to place the new hearcon immediately in its target position.

Deleting a hearcon

To delete a hearcon, it must be selected and marked before the function is evoked:

- The first alternative is to move the hearcon physically, for example, to a trashcan, and
- the second alternative is a hotkey to envoke the function.

The shrinking of a hearcon is of no special use in this approach, for the hearcons only represent the objects and do not give any information about the content of the object. Enlarging a hearcon has only the effect of enlarging the spatial boundaries in which it can be selected by the mouse cursor.

4. Restrictions for this approach

We are aware of a severe restriction for our approach. The reader may therefore not follow the impression that a GUI can directly be adapted by attaching active hearcons to all objects and thus the problems for blind users are solved. A severe problem is - certainly besides others - the number of hearcons sounding in parallel.

4.1 Simultaneously sounding hearcons

Human acoustical perception is not able to process as much information consciously in parallel as visual. As a result much fewer active hearcons may sound in order to be consciously perceived at the same time than visual icons can be shown and seen on a screen. If too many hearcons sound it is

We investigate the following solutions:

1. Not all hearcons are active objects and sound permanently but, only those hearcons close to the mouse cursor (cf. Gaver's solution in the SonicFinder, including the *soundholders*).
2. Homogenous hearcons are grouped together so that they sound as one hearcon. Auditory grouping of sounds can be implemented with similar sounds (similar timbre) or with tones which are close to one another (in frequency) [Bregman90].
3. Hearcons can be freely grouped by the user and groups can be muted ("sent to the background").

The use of tree-like structures for ordering objects (cf. the Mercator-project [Mynatt92]) is not further pursued, because they leave the spatial metaphore.

The disadvantage of the first solution is, that the user cannot directly select all hearcons, because s/he cannot hear them all. If a hearcon has to be selected, which cannot be heard, the user either knows, where the hearcon is placed, and so s/he can move the mouse in this direction. Or in the worst case the whole 3D-space has to be searched until the target object is found. A pre-condition of solution 2 is the existence of a sufficient number of similar objects, which will be presented by similar hearcons and which can therefore be grouped. The third solution allows the user to freely attach semantical group information, but the grouping is an additional activity.

Experiments will give information about how many hearcons can sound in parallel and which of the solutions is best suited when many hearcons are active at the same time.

4.2 Conclusions for this approach

As a conclusion we can state, that the hearcons and the mentioned ear-hand-coordination solutions are not absolutely qualified for adapting given graphical user interfaces, because of the problems which arise, when too many hearcons are presented at the same time. It is an open question, if any adaptation will ever enable blind people to utilize all benefits of the GUIs. Therefore in the next chapter the StereoPhonic User Interface for the Blind (SPUI-B), an assistive interface, is proposed, which offers the mentioned advantages of window systems and auditory direct manipulation for blind users.

5. SPUI-B - an assistive technology

The question which arises at this point is: why should a new assistive interface be developed especially for blind users, since it is commonly assumed that users with diasbilities are mostly interested in adaptations of programs for non- (or less-) impaired users. Our answer is, that adaptations of given GUIs will always be accompanied with a loss of information (e.g. the loss of the information, which is hidden in the topograhical arrangement of the objects). An assistive approach may offer a greater chance to build specialized interfaces for the blind in order to enable them to access topographically or topologically coded information and to utilize the functional advantages as the sighted. Besides it will be possible to represent conventional GUIs within this assistive interface, so that blind users will also be able to collaborate with sighted users in front of one screen.

5.1 Goals

The main goal of SPUI-B is to provide the advantages of the window systems for blind users, acustumed environments, secondly to control several applications in parallel by easy switching between them without great effort, thirdly the direct manipulation of hearcons (and windows) to adapt the interface to the user's requirements, and fourthly by these means to attach semantics to the location of acoustical objects in display space.

5.2 Design Concept

The basis of SPUI-B is a window system which allows users to create any number of windows, in which any application can be started, just as in visual window systems. One window is the actual window which accepts the keystrokes and which sends its outputs to the chosen output device. The sighted users will certainly prefer the screen, while blind users will use the Braille-line or the speech synthesizer. Therefore SPUI-B can be viewed not only as an special assistive program for blind persons but is also suited for sighted users[3]. Each non-actual window is represented by a hearcon. SPUI-B allows the choice of every kind of sound, music or noise or whatsoever, because a consequent limitation to one soundclass will not always be the best solution. As output technology we use the headrelated stereophony. The movement of the mouse is controlled by the ear-hand-coordination. Thus the user has the possibility to manipulate the hearcons directly as previously described. The hearcons can be placed anywhere in the acoustical space, so that the topographical arrangement is now significant for the blind user, because s/he can manipulate it.

Furthermore the user can select the hearcons directly because the position of the hearcons can be localized. As a result a switch between the applications is implemented without any great effort for the blind, who will have to select a non-actual hearcon and activate it.

A hearcon acts as a reminder, too, because it can be heard when an application is still running and has not been completed yet. To apply the marker-function equivalently to applications in a graphical window system, it must be heard *which* application is still running and not only *that* any of the applications is running (see 5.3). Again the usability depends very much on the sounds attached to the respective window and on the application.

5.3 The sound of hearcons

As already mentioned SPUI-B is not committed to any specific class for hearcons. If there is a related object in 'real life' with a typical sound, it will be represented by this sound, as far as it suits the user interaction with the program. The idea behind this approach is, that these auditory icons already give the user hints about the represented object and their semantics must not be learned. The problem is, that many objects in the computer world do not have such a relation to real objects or those real objects have no acceptable sound representations. In those cases the meaning of the hearcons must be learnt.

The use of synthetic speech for hearcons will be reduced to only occur on demand because on the one hand the hearcons are active objects which sound permanently and it will be very unpleasant to hear one word or sentence repeatedly. On the other hand speech is a rather slow output medium.

Some demands for the sounds have been stated by Mynatt [Mynatt94]:

- The conceptual mapping: Are the object represented in an obvious way.
- The user preference: Is the sound pleasant to hear.
- The physical parameters: The sounds should not mask each other, have a good sound quality etc.

Furthermore it should be possible to represent related objects by small differences in their acoustical attributes, so that their relation is perceivable.

From these demands we conclude, that hearcons can only serve as markers, when a conceptual mapping from the application to the acoustical representation has been successful. Otherwise, if the applications or objects are not acoustically represented in an obvious way, the user has to learn the meaning of the hearcons. However this contradicts the demand for an intuitive interface.

Which sounds satisfy the demands optimally, depends on the respective application and on the user. For instance, a musician will hear the hearcons in another way than a non-musician. Therefore the users must have the possibility to select appropriate sounds.

5.4 Implementation

The project has begun in 1993 and is still in the phase of fundamental research. As a technical platform we use DOS-compatible PCs enhanced with spatial sound cards and Braille-lines.

6. Conclusions

On the basis of the discussion of the pixel barrier, the mouse barrier and the graphics barrier it was shown, that some of the advantages of the graphical user interfaces cannot be utilized by blind users in an effective way. The graphics barrier ist the most difficult problem and leads to the apprehension that blind people will never be enabled to use the GUIs in way the sighted do.

Given adaptations of GUIs overcome the pixel barrier, but working with the GUIs is reduced to selecting objects and functions by using the cursor keys. The topographical arrangement of the objects stays unperceivable for the blind, so possible information coding is hidden to them. A solution to an effective mouse manipulation by blind users is given by acoustically active hearcons and the ear-hand-coordination supported by an acoustical feedback about the cursor position. Some functions to manipulate the hearcons with the mouse have been explained.

Problems of our approach are the selection of the sounds to effectively represent objects and the number of simultaneously sounding hearcons.

From the general considerations we have derived the outline of an assistive approach instead of merely adapting graphical user interfaces. This assistive approach leads to the design of the StereoPhonic User Interface for the Blind (SPUI-B).

7. Acknowledgements

The authors owe thanks to the members of post-graduate group on Psychoacoustics at the University of Oldenburg, especially to Prof. Dr. Volker Mellert and his team for their patience and their contributions in many discussions.

References

[Blattner89] M. M. Blattner, D. A. Sumikawa, R. M. Greenberg: *Earcons and Icons: Their Structure and Common Design Principles,* Human Computer Interaction 1989, Vol.4

[Boyd91] L. H. Boyd, W. L. Boyd, G. C. Vanderheiden: *Graphics-Based Computers and the Blind: Riding the Tides of Change* , in: Proceedings of the 6th Annual Conference "Technology and Persons with Disabilities" Los Angeles, 20-23.3.1991

[Bregman90] A. S. Bregman: *Auditory Scene Analysis - The Perceptual Organization of Sound* , MIT Press, Cambridge 1990

[Cohen 91] M. Cohen, L. Ludwig: *Multidimansional audio window management* , International Journal of Man-Machine Studies, 1991, Vol 34, pp.319-336

[Cohen 93] M. Cohen: *Throwing, pitching and catching sound: audio windowing models and modes* , International Journal on Man-Machine-Studies, 1993, Vol 39, pp 269-304

[Edwards88] A. D. N. Edwards: *Design of Auditory Interfaces for Visually Disabled Users* , in *Human Factors in Computing Systems* Proceedings of the CHI '88, ACM SIGCHI

[Edwards89] A.D.N. Edwards: *Soundtrack: An Auditory Interface for Blind Users* , Human Computer Interaction 1989, Vol.4

[Fricke92] J. Fricke, H. Bähring: *A graphic input/output tablet for blind computer users* , in W. Zagler (Ed.): *Computers for handicapped Persons* , Proceedings of the 3rd International Conference, Vienna, July 7-9, 1992

[Gaver89] W. W. Gaver: *The SonicFinder: An Interface that uses Auditory Icons* , Human Computer Interaction 1989, Vol.4

[Gaver90] W. W. Gaver, R .B. Smith: *Auditory Icons in Large-Scale-Collaborative Environments* , In: H. Diaper et al. (Eds.) *Human Computer Interaction -- INTERACT '90* , Proceedings of the IFIP TC 13 3rd International Conference on Human-Computer Interaction, Cambridge, U.K.. 27-31 August, 1990

[Gaver91] W. W. Gaver, R. B. Smith, T. O´Shea: *Effective Sounds in complex Systems: The Arkola Simulation* , in ´Reaching through technology´, CHI ´91 Proceedings, 1991

[GUIB93] P. L. Emiliani (Project Manager): *Publications from the TIDE project GUIB until to September 1993* , GUIB Consortium, 1993

[Mynatt92] E. D. Mynatt, W.K. Edwards: *Mapping GUIs to Auditory Interfaces* . In: Proceedings of the UIST´92, Nov.15-18, 1992, Monterey, CA

[Mynatt94] E. D. Mynatt: *Designing with Auditory Icons: How Well Do We Identify Auditory Cues* . In: Proceedings of the CHI'94 Conference, Boston, M., April 24-26, 1994

[Roederer77] J. G. Roederer: *Physikalische und psychoakustische Grundlagen der Musik* , Springer Verlag Berlin Heidelberg New York, 1977

[Schwerdtfeger91] R.S. Schwerdtfeger: *Making the GUI Talk* , BYTE, December 1991

[Smith90] S. Smith, R.D. Bergeron, G.G. Grinstein: *Stereophonic and surface sound generation for exploratory Data Analysis* in CHI ´90 Proceedings April 1990