CaMus: Live Music Performance using Camera Phones and Visual Grid Tracking

Michael Rohs
Deutsche Telekom Laboratories,
TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
michael.rohs@telekom.de

Georg Essl
Deutsche Telekom Laboratories,
TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
georg.essl@telekom.de

Martin Roth
Deutsche Telekom Laboratories,
TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
martin.roth02@telekom.de

ABSTRACT

We demonstrate that mobile phones can be used as an actively oriented handheld musical performance device. To achieve this we use a visual tracking system of a camera phone. Motion in the plane, relative to movable targets, rotation and distance to the plane can be used to drive MIDI enabled sound generation software or hardware. Mobile camera phones are widely available technology and we hope to find ways to make them into viable and widely used performance devices.

1. INTRODUCTION

Mobile phones and mobile handheld devices are by now a ubiquitous commodity used by many. They have become an integral part of everyday interactions. In this work we consider the idea of making mobile handheld technologies into new interfaces for musical expression. Many people already own mobile technologies. Hence, there is no problem of dissemination and social acceptance. Many novel interfaces for musical expression suffer from this problem because they require new and unfamiliar hardware.

Technologically mobile devices are also converging towards one entity. With respect to music this has so far been with an emphasis on playback [4, 7, 20, 26, 27, 28], and mobile music interactions with a strong input component are yet in their infancy [19]. An example of an interaction paradigm based on mobile technology is GpsTunes [25] where walking navigation is supported by variation in musical playback, which in turn is closely related to the Sonic City project, which however doesn't use a mobile device for sensing [12]. "miniMIXA" from SSEYO (www.sseyo.com) is a music mixer for mobile devices. However, it does not use camera-based input.

This work is based on continuous visual tracking and hence requires the mobile device to contain a camera with reasonable frame rates [1, 2, 23, 24]. By moving the camera phone over visual markers, the user has various degrees of freedom to control parameters which in turn are sent to a computer via Bluetooth. The incoming signal is converted to MIDI messages, which then can be mapped to MIDI-enabled sound generating software and hardware.

NIME 06, June 4-8, 2006, Paris, France

The visual display of the phone is used to provide navigational guidance and allow for authoring of the geometric setup of the mapping. A flow diagram of the system is shown in Figure 1.

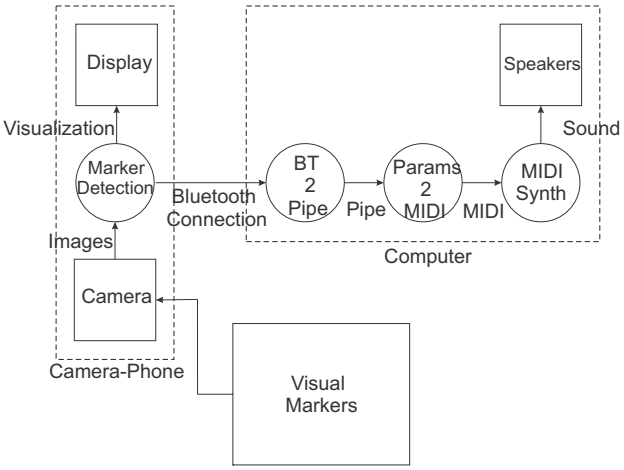


Figure 1: Overall flow diagram of the interface design. The camera phone is used to continuously sense a visual marker grid and display guidance whereas a computer is used to map the data to MIDI and in turn produce sound.

We use a commercial music sequencing program to demonstrate the potential of the mobile interaction for live performance. For this purpose a piece has been written by the third author which can then be remixed live using the CaMus technology.

Other projects have used visual marker recognition to devise new interfaces for musical expression [5, 10, 18]. In all these cases a stationary digital camera is connected directly to a computer and mobile camera phones are not used. Visual tracking for physical mobile interaction has been described in [13, 14, 23]. Hansen et al. [14, 15] track hand-drawn circles, colored objects, and faces, in order to implement *mixed interaction spaces*: physical spaces in which digital interaction takes place. Hachet et al. [13] propose a camera-based interface for two-handed input. One hand holds the device, the other hand holds a card with color codes. The 3D position of the camera relative to the card is mapped to different user interface operations such as pan and zoom, and rotation, and navigation in tree maps.

Figure 2: The user focuses a target that is located at a fixed position in the virtual workspace.

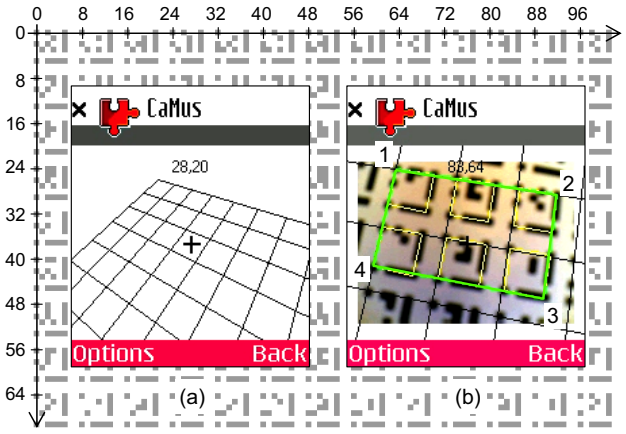


Figure 3: The grid defines an absolute coordinate system. The camera phone computes the coordinates of the cross hair (28,20 in the left screenshot), the distance from the grid surface, and the amount of rotation and tilting. (a) The grid is drawn according to the current camera perspective. (b) The perspective mapping is computed from the maximum area of recognized markers.

2. VISUAL GRID TRACKING AND PERSPECTIVE RENDERING

Interaction takes place on a grid of visual markers, which are derived from visual codes [23]. The grid represents a large workspace, of which different parts can be accessed with a camera phone by simply placing it over the relevant area (see Figure 2). The phone display acts as a window into the virtual workspace. The grid thus enables a spatially aware display [11, 29]. We present one-handed techniques for interactions with objects in space. The space can be shared by multiple camera phones.

The grid defines a coordinate system that provides an absolute frame of reference for the spatial interaction (see Figure 3). Printed on paper, it typically extends over a DIN A4 or A3 sheet. The upper left corner of the grid is the origin, the upper edge is the x-axis, and the left edge is the y-axis of the grid coordinate system. One coordinate unit corresponds to a single black-and-white cell. Each marker has a width and height of 6 cells. Markers are placed two coordinate units apart, which results in one marker for each 8×8 unit area of the grid. The left upper corner stones of each marker are placed at grid coordinates $(8x, 8y), x, y \in \{0, 1, 2, ..., 31\}$.

The markers have a layout similar to visual codes [23], but consist of only two corner stones and two guide bars and a smaller data area. The data area of a single marker has a raw capacity of 12 bits. It is used to store the x index (5 bits) and the y index (5 bits) of the marker within the grid, as well as two parity bits. A grid can thus have a maximum size of 32×32 markers, which is equivalent to 256×256 coordinate units.

In our current implementation on Symbian camera phones with a resolution in view finder mode of 160×120 pixels and with a printed size of 6.8 grid units per cm (i.e. the size of a single marker is about 9×9 mm), the grid is detectable at distances between 2 cm and 10 cm from grid surface to camera lens. This results in a 3D interaction space of *length* × *width* × *height* = 37.6 cm × 37.6 cm × 8 cm, if the maximum of 32×32 markers are used. In this space, we can precisely determine the position and orientation of the phone at a rate of up to 10 frames per second with our prototype device (a Nokia 6630 Symbian phone), depending on the complexity of the rendered virtual workspace. In particular, the focus point on the grid surface can be tracked with high precision.

The grid has to allow for a smooth continuous detection of position and orientation during movement. It also has to provide the basis for a perspective rendering of the grid lines as shown in the left screenshot in Figure 3 (labeled *a*). (Here, the distance between parallel lines is 8 units in the grid coordinate system.) Perspective rendering makes the illusion more convincing, as if the user is looking "through" the device screen onto the background. This is similar to the effect of *symbolic magnifying glasses* [22] and *see-through tools* [6]. To fulfill the first requirement of continuous grid tracking, we have devised a very small visual marker, such that at any position on the surface, and even at close camera distances from the grid, there is at least a single visual marker completely contained in the camera image. However, with smaller sized codes, the perspective mapping between corner points of the code and corresponding grid coordinates is less accurate. Thus, in order to fulfill the second requirement of providing a stable basis for perspective rendering, we use multiple markers in the camera image to establish the mapping. This is shown in the right screenshot in Figure 3 (labeled *b*).

We need four corresponding pairs of (triple-wise non-collinear) points to establish a perspective mapping between the image coordinate system and the grid coordinate system. Once the markers are recognized, the image pixel coordinates of their corner stones and guide bars are known. This is indicated by the yellow frames around each recognized marker in Figure 3b. (The camera image is not shown during normal use, since it distracts the user.) The closer these points lie together, the less accurate the mapping will be, since a small variation in corner stone pixel positions of successive camera frames results in severe fluctuations of the grid rendering. However, at medium distances, multiple markers are present in the camera image. Hence, to compute the perspective mapping we use the largest possible area to get the most accurate mapping.

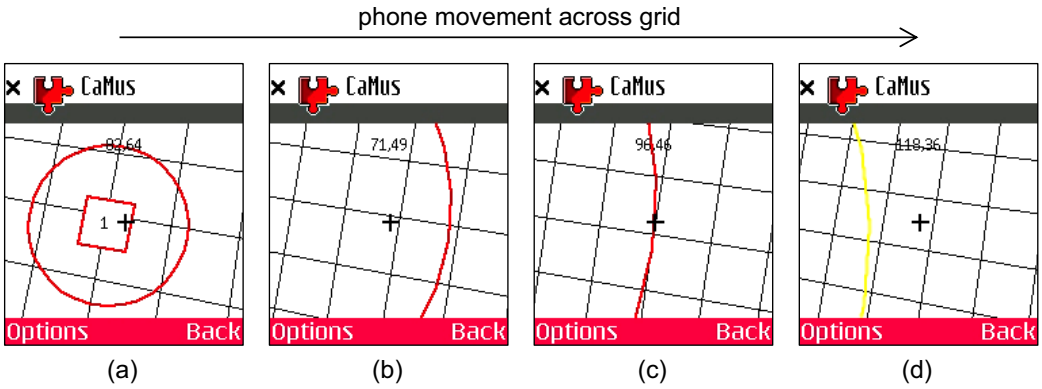


Figure 4: A halo is used to indicate a distant target. Moving from left to right with increasing distance from the target: (a) target is focused, (b) inside the target's range of influence, (c) on the border of the range of influence, (d) outside the range of influence (color changes from red to yellow).

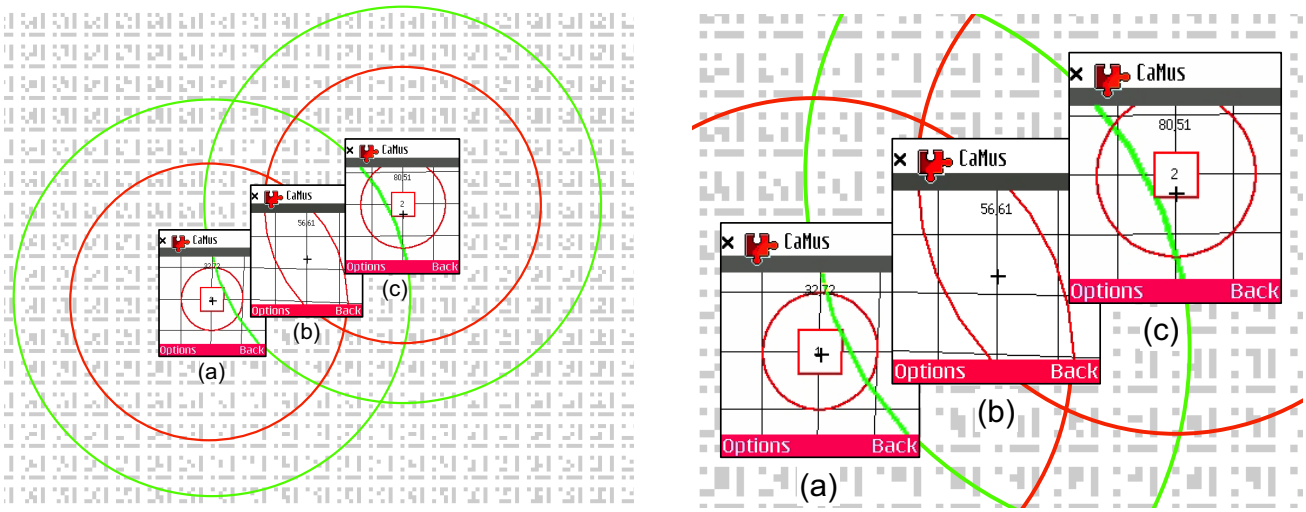


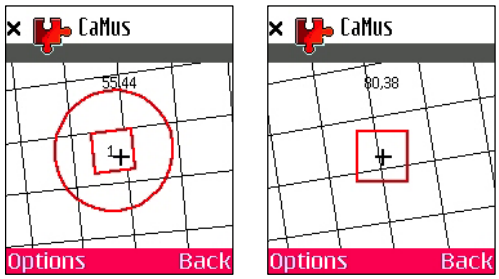
Figure 5: Screen contents at three different positions on a grid with two targets (the right half is a magnification of the left center area): (a) left target focused, (b) between the two targets, (c) right target focused.

In the frame shown in Figure 3b, six markers have been detected. They are highlighted by yellow frames. Instead of basing the perspective mapping on any of the markers' corner points, the points (1) to (4) are used, which represent elements of different markers. The resulting rectangle is highlighted by a green frame. For these corner points, of which we know the image coordinates from the marker recognition step, we establish correspondences as follows. For an upper left corner, its grid coordinates are $(8i_1, 8j_1)$, where i_1 and j_1 are the horizontal and vertical indices that are stored in the marker. The grid coordinates of an upper right corner are $(8i_2 + 5, 8j_2)$, of a lower right corner $(8i_3 + 5, 8j_3 + 5)$, and of a lower left corner $(8i_4, 8j_4 + 5)$. Again, i_k and j_k, $k \in \{1, 2, 3, 4\}$, are the vertical and horizontal index values that are stored in the respective code. From these correspondences, we can compute a perspective mapping (a planar homography) between image coordinate system and grid coordinate system. With this mapping and its inverse, graphical elements that are specified with respect to the grid coordinate system are translated to image coordinates and focus point coordinates are translated to the corresponding point on the grid.

3. VISUAL GUIDANCE

We use the term *target* to denote the basic elements that can be created and placed in the workspace. Each target has a position in grid coordinates and an index number associated to it. Since the small display of a mobile phone can only visualize a small part of the workspace at once, finding targets at different positions can quickly become difficult. Since we want to avoid the need for switching between an overview mode and a closeup mode, we decided to use an extension of the *halo* technique [3] as a way to visualize off-screen targets. This technique supports spatial cognition by surrounding targets with rings that are just large enough to reach into the display window (see Figures 4 and 5). Even if the visible arc is only a small fraction of the ring, it contains all the information needed to intuitively infer the direction and approximate distance of a target. The technique uses very little screen space and has been shown to lead to significantly shorter task completion times compared to arrow-based visualization techniques [3].

Our extension relates to the visualization of the *radius of influence* – and equivalently the circular *area of influence* – that is associated to each target. If the user focuses



a) focus target, press button

b) drag target, while button pressed

Figure 6: Repositioning targets on the grid: (a) focus target and press joystick button, (b) drag target to new location and release joystick button.

a grid point outside the area of influence, then the target has no effect on the output produced. If a point inside the area of influence is focused, then the strength of the effect is inversely proportional to the distance from the center. The goal was to find a visualization that uses the halo technique and does not add additional visual clutter to the interface, yet intuitively tells the user if he is inside or outside the area of influence. Figure 4 shows how the screen contents change as the user moves to the right, starting from the center of a target (a), at which it is surrounded by a minimal halo. Moving to the right, but still inside the area of influence (b), the halo remains to the right of the cursor. If the cursor is inside the arc of the halo, this indicates "insideness." Around the radius of influence (c) there is a boundary region, within which the halo is drawn at that radius at a constant position with respect to the background grid. With further movement, the halo moves with the background grid to the left. When it passes the cursor, its color changes from red to yellow. Color and curvature of the halo now indicate that the user is located outside the area of influence. Using this extension we have reached multiple goals. With a minimal amount of visual clutter, we visualize "insideness" and "outsideness" and the direction and approximate distance of off-screen targets.

The technique scales to multiple targets, as well. Figure 5 shows a grid with two targets. Target 1 is located at grid coordinates $(32, 72)$ and target 2 at $(80, 48)$. The targets are located just outside each other's radius of influence. At position (a), target 1 is focused and the outside of target 2's halo is visible. At position (c) it is the other way round. At the intermediary position (b), the cursor is inside both areas of influence, as indicated by the halos. As proposed in [3], if many targets are present, it is advantageous to merge multiple halos into a single one to indicate a cluster of distant targets.

There are a number of different possibilities of interacting with a target. First, targets can be created at the current cursor position by pressing the phone's binary joystick (also termed multi-way button) in south direction. Pressing it in north direction deletes the closest target.

Targets can be also be freely repositioned on the grid (see Figure 6): (a) a user focuses the target and presses the joystick button, (b) the minimal halo disappears and the target can be dragged to a new location. The target follows the cursor until the button is released.

4. INTERACTION TYPES

Given this technology a number of parameters can be detected. First the x and y position along the two orthogonal axes of the visual marker array. Second is the rotation of the device with respect to the visual marker panel. Distance to the marker array can also be extracted including tilt. In our current implementation we only use position, rotation and distance. This information is then sent to software on the laptop for mapping to MIDI. At this stage already some transformations are applied. In our current implementation, xy-coordinates are converted into relative distances to active markers (see section 3) with a maximum radius of influence, height, and rotational angle in the range of -90 to 90 degrees with 0 degrees being north. Then the respective parameters are linearly rescaled to fit the typical dynamic range of MIDI (0-127).

This information is only sent while the cursor is in the influence radius of a target. In this case rotation of the phone in the horizontal plane 90 degrees left and right of the normal upward position and height of the phone over the sheet with the visual marker array will be sent and are associated to MIDI messages.

An important result is the ability to easily author and perform "cross-mixed" effects. Targets can be placed in relative position to each other. The distance to each other will define how much the respective effects overlap in various spatial regions. By moving inside and outside of these regions effects can be manipulated individually and jointly in a continuous and intuitive way.

An existing target can always be picked up and moved by use of the joystick button. This allows a change in the current effects cross-mix. If a target is no longer desired it can also be removed.

It is important to note that these choices are already arbitrary. Alternative designs are thinkable. For example, the x and the y axis can be used separately to implement axially dependent controls, for example up-down controlling one parameter like a vertical slider and left-right doing the same for a horizontal slider.

Our subjective experience suggests, that the notion of distance is rather intuitive. It is not prone to problems due to a difference in alignment of hand motion direction to marker sheet axis.

Our current interaction mechanisms have two target audiences in mind. First a knowledgeable performer, who is aware of the relative effects of mapped targets. This user can place targets and move them into relative position to compose a parameter landscape. The second audience is a novice user, who can explore this landscape with minimal musical knowledge and achieve musically interesting results by exploration.

5. MAPPING

The targets themselves do not offer an inherent semantics for their use in interaction. The mapping of data associated to them (position, rotation and height) can be freely mapped to sound synthesis parameters, effects parameters or other attributes chosen for control. This is a source of great flexibility of this implementation but at the same time also brings this device close towards the classical mapping problem of new musical instruments where a gesture may not have an intrinsic musical meaning [17].

In our example mappings we chose to map height to overall strength of an effect, relative distance in the horizontal plane to a target for a primary effect parameter, which results in controlling the most salient feature of the

effect. If a secondary effects parameter was of interest it was mapped to rotation of the device in the horizontal plane. The first two gestures are similar to the analogy of gravity or strength of influence by proximity, whereas the last gesture resembles the gesture used for knob or dial control.

6. SOUND

Figure 7: An example mapping of CaMus to the commercial sequencing software FruityLoops (by Image-Line Software). Various parameters are mapped to the knobs of various effects boxes via MIDI.

The interface can be used with any software that offers MIDI input. We experimented with the commercial sequencing software FruityLoops (by Image-Line Software, see Figure 7), as well as the research projects STK [8, 9] and PD [21].

Our main sounding environment and application has been on the fly re-mixing of music. The third author wrote a piece within FruityLoops for this purpose. The ability of FruityLoops to link control-knobs of its interface to incoming MIDI messages was then used to map the controls of the camera phone to various pre-selected effects (Figure 7). The respective mappings are depicted in Table 1.

6.1 Sound, Gesture and Visual Display

For each of the effects, a separate target can be freely placed within the plane. The performance consists of two basic features.

The first is the authoring or configuration feature. In this case, targets are placed and moved in the plane. This happens interactively by clicking a button on the camera phone to either place a new one or grab an existing one to drag it. Dragging continues as long as the button is held. As long as a target is present it emits control signals to the laptop and in turn to the MIDI software. The main function of this step is to pick effects which are desired for a performance. One can either fix their relative spatial location before the performance starts or change the relative positions between targets on the fly during a performance.

The second feature consists of performing with one or more interaction target. Whether or not a target is currently being a source of manipulation is determined by the distance of the camera cursor position relative to the target position. If the cursor is within a preset range, all effects of that target will be active and sent to the MIDI

Effect	Distance	Height	Rotation
Distortion	Distortion	Effect weight	Not used
LP filter	Cut-off freq.	Effect weight	Not used
Balance	Not used	Effect weight	Left/right
Delay	Forward delay	Effect weight	Feedback delay
Reverb	Reverb	Effect weight	Room size

Table 1: Mapping of CaMus parameters to digital audio effects.

software. Because multiple targets can be within range of the current cursor position, multiple effects of the MIDI software can be manipulated simultaneously. At the same time, their respective spatial separation allows for various degrees of influence or configuration of the effect leading to a sort of multi-effect hybridization. For example, one can place two targets with some separation. In the region between the targets both effects will receive high control values, whereas in the regions on one side or the other of the target leads to a joint effect where one effect parameter is low while the other parameter is high. This joint parameter space can be conveniently explored by moving in the plane and easily extends to multiple effects (see Figure 5). The behavior of the effect is intuitive by the "strength of proximity"-analogy.

The result is an easy to use multi-effects performance interface, which in this particular setting can be used for life remixing of sequenced software using real-time digital effects.

7. CONCLUSIONS

We presented a mobile camera phone based interface for musical expression, which uses visual marker technology to allow hand gesture based performance of music and music remixing.

The main motivation behind developing this technology is a first indication at the potential of commodity handheld mobile devices serving as novel interfaces for musical expression. These devices are already very widely available and hence one can hope that expressive musical uses can more readily reach a large practitioner base.

We have described a camera based sensing technology that was used to design interactions for musical performance allowing for motion in the plane, hand rotations in the plane as well as distance control. By using MIDI to interface with the synthesis software, a large array of sounding sources and specific mappings can be achieved.

Future plans include various modes of collaborations, including the use of phones interacting with each other, the extension of the plain target paradigm to allow a variety of analogies, potentially including sliders and dials. We are currently to implementing a synthesis engine on the mobile phone to allow stand-alone use of the mobile phone for performance. Finally we plan to either augment or substitute the current sensing technology with other modes of sensation [16] and provide additional sensory modes of display, for example tactile.

In terms of camera-based input, one future direction is tracking everyday types of visuals – like faces, posters, or specific colors – and map them to characteristic musical output. Another direction is to capture phone movement with optical flow techniques and map it to filters. This could also be the basis for the recognition of compound gestures that are linked to specific musical output.

8. REFERENCES

[1] R. Ballagas, M. Rohs, and J. G. Sheridan. Sweep and Point & Shoot: Phonecam-Based Interactions for Large Public Displays. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1200–1203, New York, NY, USA, April 2005. ACM Press.

[2] R. Ballagas, M. Rohs, J. G. Sheridan, and J. Borchers. The Smart Phone: A Ubiquitous Input Device. *IEEE Pervasive Computing*, 5(1):10–17, Jan. 2006.

[3] P. Baudisch and R. Rosenholtz. Halo: A Technique for Visualizing Off-Screen Objects. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 481–488, New York, NY, USA, 2003. ACM Press.

[4] F. Behrendt. *Handymusik. Klangkunst und 'mobile devices'*. Epos, 2005. Available online at: `http://www.epos.uos.de/music/templates/buch.php?id=57`.

[5] R. Berry, M. Makino, N. Hikawa, and M. Suzuki. The Augmented Composer Project: The Music Table. In *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 03)*. IEEE, 2003.

[6] E. A. Bier, M. C. Stone, K. Fishkin, W. Buxton, and T. Baudel. A Taxonomy of See-Through Tools. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 358–364. ACM Press, 1994.

[7] W. Carter and L. S. Liu. Location33: A Mobile Musical. In *Proceedings of the Mobile Music Workshop*, Vancouver, Canada, May 25 2005.

[8] P. Cook and G. Scavone. The Synthesis ToolKit (STK). In *Proceedings of the International Computer Music Conference*, Beijing, 1999.

[9] P. R. Cook. *Real Sound Synthesis for Interactive Applications*. A K Peters, Ltd., July 2002.

[10] E. Costanza, S. Shelley, and J. Robinson. Introducing Audio D-Touch: A Tangible User Interface for Music Composition and Performance. In *Proceedings of the International Conference on Digital Audio Effects (DaFX-03)*, 2003.

[11] G. W. Fitzmaurice. Situated Information Spaces and Spatially Aware Palmtop Computers. *Commun. ACM*, 36(7):39–49, 1993.

[12] L. Gaye, R. Mazé, and L. E. Holmquist. Sonic City: The Urban Environment as a Musical Interface. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Montreal, Canada, 2003.

[13] M. Hachet, J. Pouderoux, and P. Guitton. A Camera-Based Interface for Interaction with Mobile Handheld Computers. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 65–72, New York, NY, USA, 2005.

[14] T. R. Hansen, E. Eriksson, and A. Lykke-Olesen. Mixed Interaction Space: Designing for Camera Based Interaction with Mobile Devices. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1933–1936, New York, NY, USA, 2005. ACM Press.

[15] T. R. Hansen, E. Eriksson, and A. Lykke-Olesen. Mixed interaction spaces – a new interaction technique for mobile devices. Demonstration at UbiComp, 2005.

[16] S. Hughes, I. Oakley, and S. O'Modhrain. MESH: Supporting Mobile Multi-modal Interfaces. In *Proceedings of UIST'04*, Sanfa Fe, NM, 2004.

[17] A. Hunt, M. M. Wanderley, and M. Paradis. The Importance of Parameter Mapping in Electronic Instrument Design. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, pages 149–154, Dublin, Ireland, May 24-26 2002.

[18] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reacTable*. In *Proceedings of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, 2005.

[19] M. Kaltenbrunner. Interactive Music for Mobile Digital Music Players. In *Inspirational Idea for the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain, 2005.

[20] G. Levin. Dialtones - A Telesymphony. Description of the Dialtones concert performances `http://www.flong.com/telesymphony/`, September 2 2001. Retrieved on January 23, 2006.

[21] M. Puckette. Pure Data. In *Proceedings of the International Computer Music Conference*, pages 269–272, San Francisco, 1996. International Computer Music Association.

[22] J. Rekimoto. NaviCam: A Magnifying Glass Approach to Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):399–412, Aug. 1997.

[23] M. Rohs. Real-World Interaction with Camera Phones. In H. Murakami, H. Nakashima, H. Tokuda, and M. Yasumura, editors, *Second International Symposium on Ubiquitous Computing Systems (UCS 2004), Revised Selected Papers*, pages 74–89, Tokyo, Japan, July 2005. LNCS 3598, Springer.

[24] M. Rohs. Visual Code Widgets for Marker-Based Interaction. In *IWSAWC'05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems – Workshops (ICDCS 2005 Workshops)*, pages 506–513, Columbus, Ohio, USA, June 2005.

[25] S. Strachan, P. Eslambolchilar, R. Murray-Smith, S. Hughes, and S. O'Modhrain. GpsTunes: Controlling Navigation via Audio Feedback. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services*, Salzburg, Austria, September 19-22 2005.

[26] N. Warren, M. Jones, S. Jones, and D. Bainbridge. Navigation via Continuously Adapted Music. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1849–1852, New York, NY, USA, 2005. ACM Press.

[27] D. Waters. Mobile Music Challenges 'iPod Age'. BBC News Online, Monday, 7 March 2005. Available online at `http://news.bbc.co.uk/1/hi/technology/4315481.stm`.

[28] T. Yamauchi and I. Toru. Mobile User-Interface For Music (poster). In *Proceedings of the Mobile Music Workshop*, Vancouver, Canada, May 25 2005.

[29] K.-P. Yee. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–8, New York, NY, USA, 2003. ACM Press.