

MITSUBISHI ELECTRIC RESEARCH LABORATORIES
<http://www.merl.com>

Parsimony & Transparency in Ubiquitous Interface Design

Christopher R. Wren and Carson J. Reynolds

TR2002-022 September 2004

Abstract

Minimalism in ubiquitous interface design allows computational augmentations to seamlessly coexist with existing artifacts and the constellations of task behaviors surrounding them. Specifically, parsimony and transparency contribute to improved learnability and user acceptance of novel interfaces.

*Appears in Personal & Ubiquitous Computing Journal:
Special Issue on Tangible Interfaces in Perspective*

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2004
201 Broadway, Cambridge, Massachusetts 02139

- September 2004:** Personal & Ubiquitous Computing Journal Special Issue on Tangible Interfaces in Perspective. 8(5) pp. 370–373. Springer-Verlag London Ltd
- September 2002:** Fourth International Conference On Ubiquitous Computing, Adjunct Proceedings. September 29, 2002, Gteborg, Sweden

1 Design Overview

Minimalism in ubiquitous interface design allows computational augmentations to coexist with unmodified artifacts and the constellations of task behaviors surrounding them. By transparently integrating aspects of the digital world into real artifacts, we strive to provide ubiquitous interfaces to computation that do not obscure or destroy the highly refined interaction modalities of everyday artifacts in the physical world.

We present a system that demonstrates this design philosophy: an augmented *go* board. The game of *go* is an intriguing test case because it is surrounded by a set of behaviors and aesthetic concerns that have been refined over the course of thousands of years. Our system provides a flexible, mode-less augmentation of *go* by adhering to minimalism. Our design philosophy conceives minimalism as a combination of parsimony and transparency.

Minimalism doesn't necessarily imply limited functionality. Transparent design means minimizing cognitive demands on the user by limiting the changes to the pre-existing constellation of behaviors surrounding the artifact being augmented. For example, the augmented *go* system transparently adds game recording and an automatic move clock to traditional face-to-face play with no change to the traditional experience.

Furthermore, parsimony means minimizing the introduction of interface elements and inappropriate metaphors that could lead to clutter. The traditional activities of solitary review, study, and problem solving are enhanced by the addition of minimal visual augmentations that are appropriate to the game context and therefore preserve the game aesthetics. The traditional experience is actually improved because the user is free not only from distractions in the interface, but also from the usual distractions of notes, reference books, and newspapers.

One aspect of this work is the constructive coexistence of the physical and the virtual. In this respect this work is similar to the work of Wellner on the DigitalDesk[15] and is informed by Ishii's pioneering efforts in tangible user interfaces[13]. Another aspect of this work is the desire for transparency and minimal cognitive demands on the user. In this respect it is inspired by work on sympathetic interfaces[8] and supported by the prior literature on perceptual interfaces[16]. In choice of domain, the augmented *go* board is most similar to Underkoffler *et al's* tangible chess board[14]. *Go's* visual affordances provide a simple domain for computer vision techniques. One shortcoming of focusing on *go* as a domain is that the application of minimalism to more complex domains was unexplored. However, after anecdotal experiences augmenting another artifact (a desktop calculator) we think minimalism is applicable to other simple domains.

As the illustrations below show, the *go* system has been implemented. Furthermore, we have conducted an evaluation of the system. An experiment conducted to assess learnability found that subjects not only preferred the minimalist tangible interface when compared to a more traditional GUI, but they also performed tasks significantly more rapidly with this interface[17].

Using minimalism as a guiding design principle suggests learnable methods for interacting with ubiquitous computers. Users of ubiquitous computers will likely have no desire to learn thousands of user interfaces for every conceivable use context. Instead, we propose designers extend a minimal design philosophy to other artifacts, allowing users to interact with them in familiar ways.

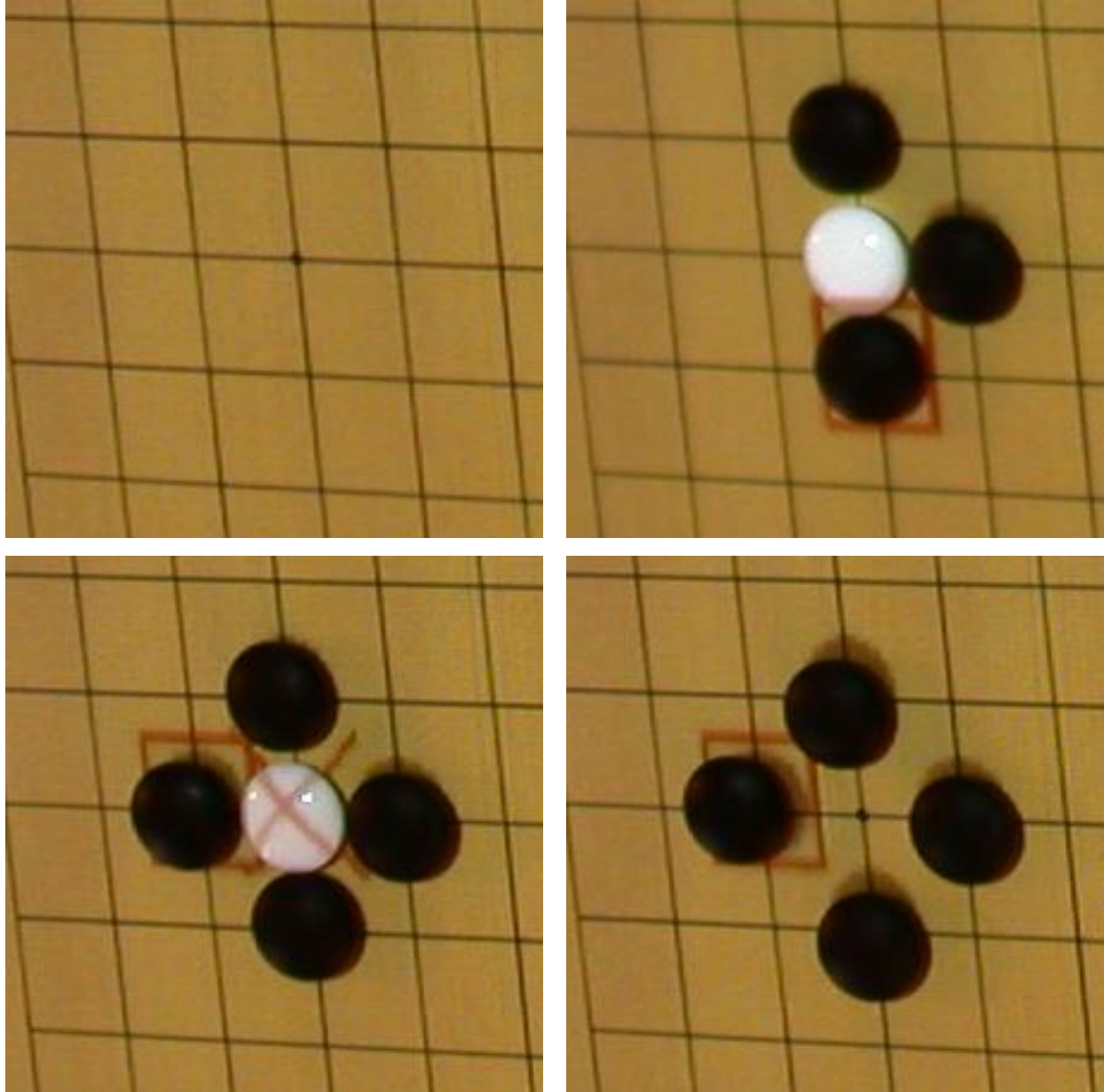


Figure 1: A series of illustrations showing some of the augmentations available to the system. They allow the system to highlight the last move (this is useful for study and remote play), inconsistent board state (useful during study, remote play, and even during live tutoring), and piece confirmation (which provides a subtle cue to the human that there is an additional computational entity that attending to the interaction).

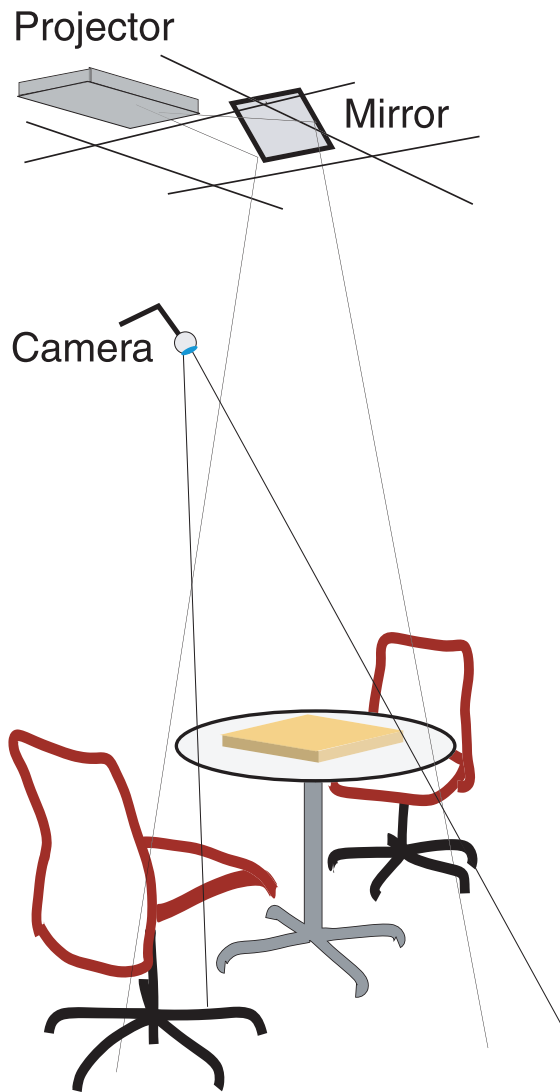


Figure 2: The system we implemented is governed by the design philosophies of parsimony and transparency articulated above. The system itself consists of a light-table comprised of a video camera and projector situated above a go board on normal table. The system projects visual annotations that form a superset of the traditional board functionality. The vision system explicitly supports our design philosophy by accommodating the traditional style of game play.

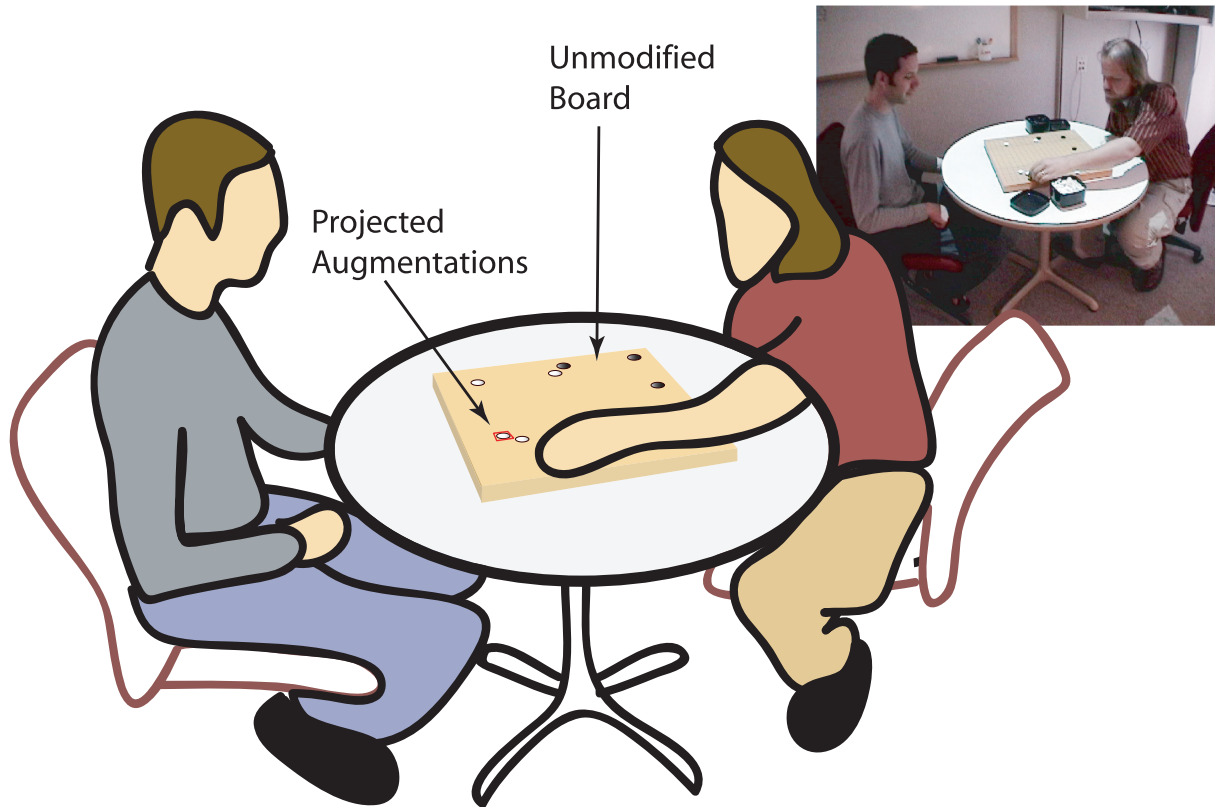


Figure 3: The players utilize the board as they always have, playing a game without having to wear any special markers, or explicitly interacting with computers, or other devices. The environment is not cluttered with virtual buttons, or menus, or other inappropriate carry-overs from the GUI domain.

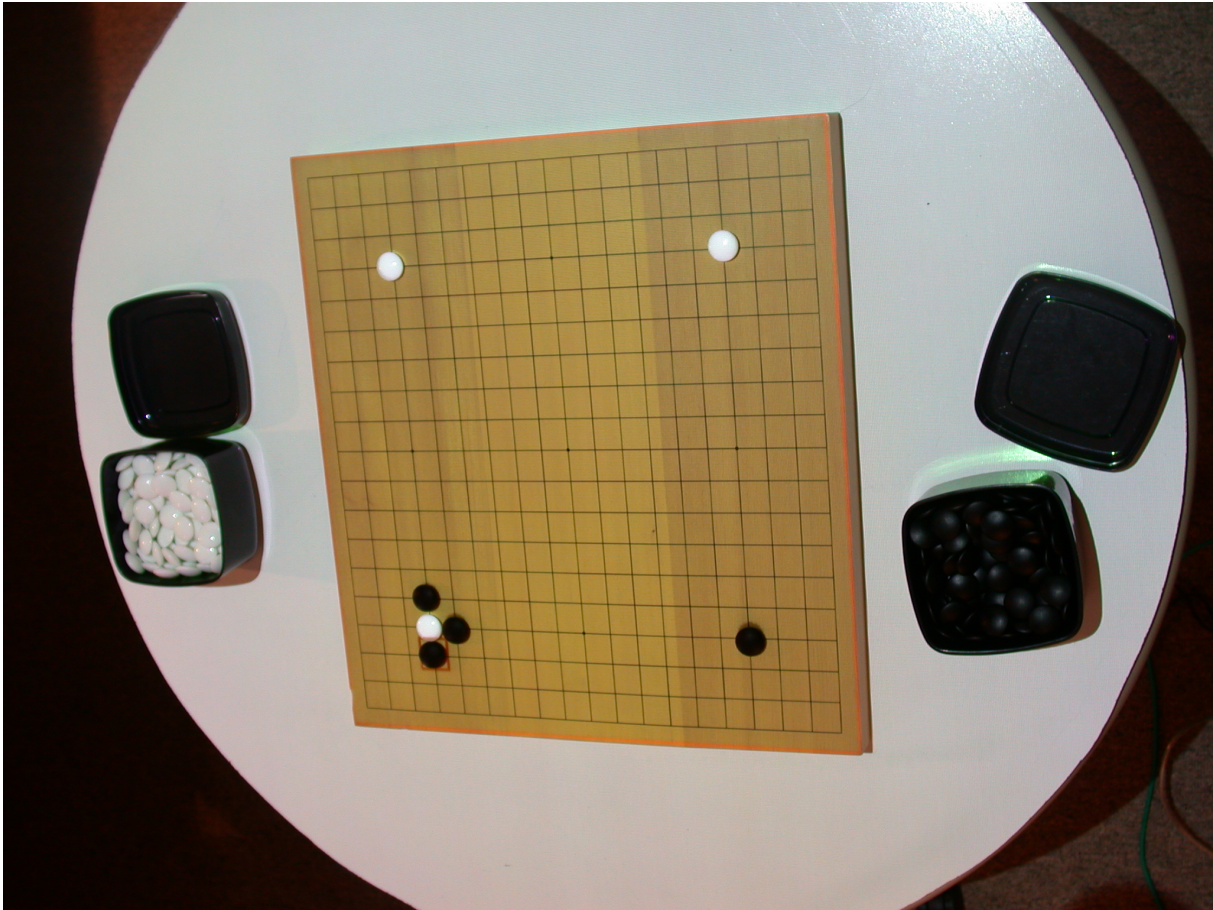


Figure 4: The view from the camera while a game is being played, between moves. The system employs well-understood computer vision techniques to recover the state of the world. A rich domain model allows the perceptual system to react intelligently to human actions, without introducing onerous constraints or confusing modal behavior. The system does not control the interaction: it only provides helpful information in response to free-form human activity.

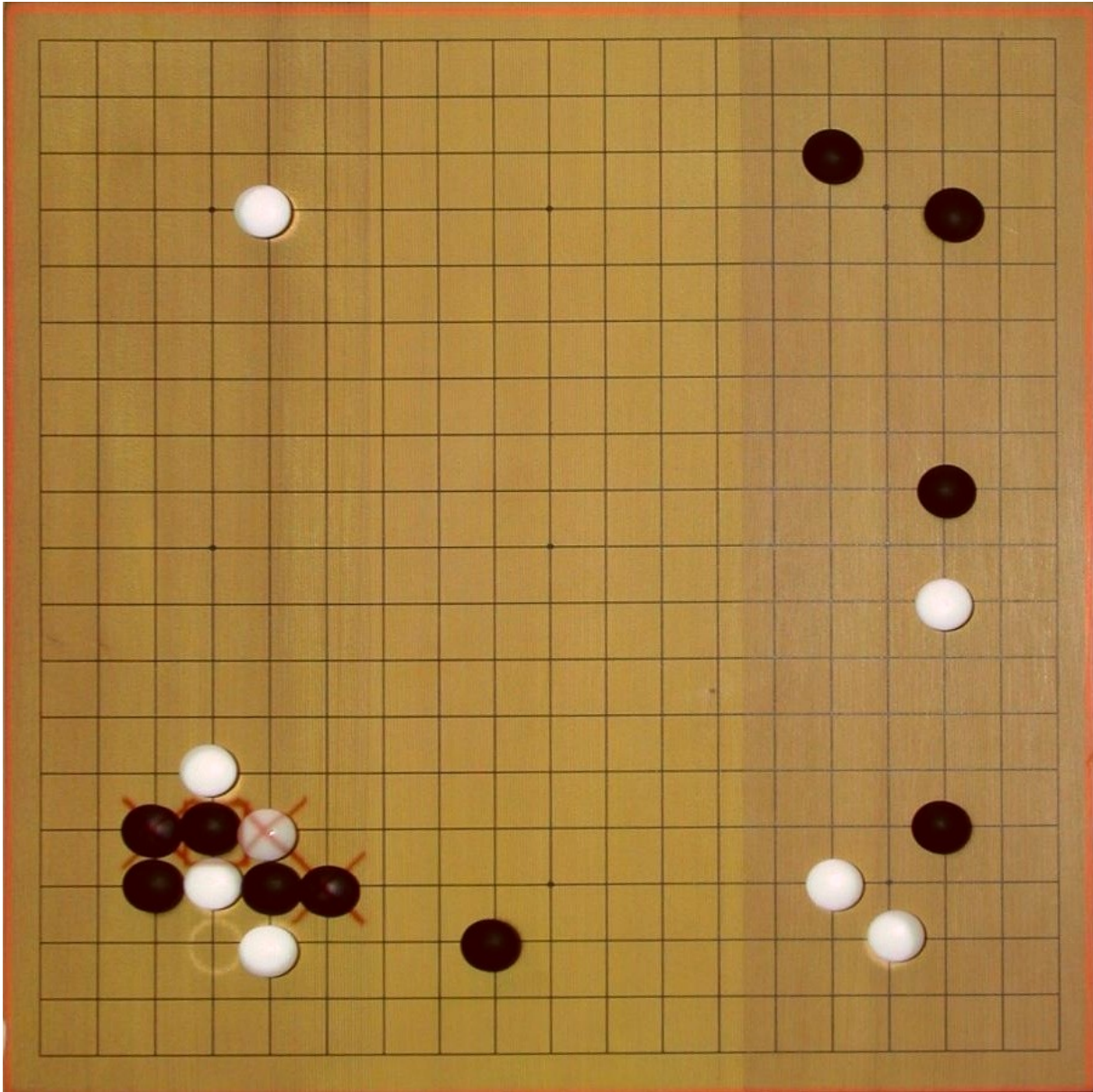


Figure 5: An illustration of the roll-back interface. The student has explored a variation on a recorded game and is attempting to go back to the primary task. The system highlights the student's options in the lower-left. This is a form of cognitive aid that is essential to novice study, and it is provided by every GUI *go* client that we surveyed, but until now, was lacking from the physical interface.

2 Tangible Intermediaries

At the highest level, we use tangible intermediaries as an organizing design principle; a tool around which interaction is structured. Our approach builds on the notion of *direct manipulation*, a widely used interaction design principle[11]. It dictates that users should interact with representations directly (using mouse and keyboard). A drawing program where users paint by dragging the mouse over virtual paper allows users to directly manipulate the drawing image. In many ways tangible intermediaries are an extension of direct manipulation, with the exception that users physically interact with the target of the manipulation.

2.1 Definition

A tangible intermediary is an existing object that is used as an organizing principle for interaction with augmented reality systems. As the name implies, the object should be both:

- **tangible**, namely physical and graspable
- **intermediary**, or conceptually situated in between the user and computation with which they are interacting.

As a principle, it prescribes that augmented user interaction with a particular physical object ought to be mediated by that very same object. It then follows the philosophical principle of parsimony, sometimes called Ockham's Razor. Specifically: "plurality is not to be posited without necessity"[2]. Our corollary is that extra interfaces should not be implemented without necessity. Or interfaces that do not make use of the augmented object are superfluous.

It should be emphasized that our definition does not extend to abstract widgets that may be coupled with digital information. Most strictly, a tangible intermediary is based on a traditional artifact and aids users in performing a task by supporting and leveraging the existing set of skills surrounding that object.

2.2 Related Approaches

Our approach is similar to the work of Pierre Wellner on the DigitalDesk. Wellner sought to equalize the disparities between the digital and physical worlds by providing a permeable membrane between them. The DigitalDesk differs from our work in its focus on paper as a medium for office tasks[15].

The work of Ishii on tangible user interfaces (TUIs) would on the surface appear to be the most strongly related research. TUIs allow the user "to grasp and manipulate [bits] . . . by coupling the bits with everyday physical objects and architectural surfaces" [6]. Ishii and his collaborators provides concrete examples of TUIS for diverse tasks like urban planning and playing ping pong, among others[14].

Conceptually, TUIs are positioned as a way to "give physical form to digital information." This focus on digital information leads Ullmer and Ishii to characterize TUIs and draw parallels to existing graphical user interface (GUI) elements. Specifically they "physically instantiate GUI 'icons' as TUI 'phicons' (physical icons) with varying levels of representational abstraction"[12]. We perceive TUIs as a broad effort to give physical form to digital, computational elements.

While these elements have their use, in the realm of ubiquitous computing, they may introduce problems. Specifically, if we imagine using computers embedded universally, then to interact with them *in situ* using phicons requires that these widgets would also have to be universal. If we further imagine that there would be innumerable computers involved in the enterprise of ubiquitous computing, then the problem begins to emerge. To outfit each of the innumerable applications of ubiquitous computing with innumerable more phicons would result in overwhelming physical clutter. Phicons don't scale to the immense task of interacting with ubiquitous computers.

Our approach deviates from this approach in some interesting ways. Tangible intermediaries are strongly influenced by the artistic and musical style of minimalism. Other researchers have applied the concept of minimalism to the design of documentation[3]. The tangible intermediary approach emphasizes a sort of interface minimalism; the interface is bounded by the affordances of the augmented object.

Johnson's sympathetic interfaces[8] share a fundamental tenet with tangible intermediaries: to designing easily learnable interfaces by significantly reducing the cognitive distance between the interface and the task. However, the primary task for the sympathetic interface is the indirect manipulation of a virtual object. The physical object is a representational tool, not necessarily the center of focus for the interaction. Similarly, the considerable work on virtual environments[4, 16, 1] focuses on creating kinesthetic interfaces to computation, but the focus of the interaction is commonly inside the computer: a virtual game, a story, or a data set. There is a layer of abstraction between the physical experience of the user and the task.

2.3 Advantages

Tangible intermediaries offer up some unique advantages over virtual interfaces. We have found that tangible interfaces are easier for novices to learn than graphical user interfaces. We think this is because users have immeasurably more experience interacting with physical objects than virtual ones.

We have also found by implementing several systems based on tangible intermediaries that it is straightforward to design interaction for them. The arduous job of modeling the task that users perform with software is made easier because physical artifacts often have a limited and readily observable set of activities that people perform with them.

Users can also interact with tangible intermediaries in a faster manner. An evaluation detailed below shows that participants can use tangible intermediaries more quickly than graphical user interface equivalents.

Other research indicates that users can perform some tasks in a better manner with tangible user interfaces than with their graphical counterparts. Patten and Ishii found experimentally that users had more success with a task involving spatial organization and recall using TUIs than users performing the same task with GUIs[9].

2.4 Limitations

Certain sorts of applications have no obvious intermediary that could be used. Abstract tasks like programming, or manipulating files would be somewhat awkward with a tangible intermediary interface. It may be more appropriate to Perform these tasks with a traditional GUI or TUI.

Using computer vision and projection is also computationally expensive. It is our assumption that soon the demands of these sort of applications will be well within the capabilities of most computers. But implementors

must bear the stringent processing requirements applications which observe tangible intermediaries require. Computationally expensive systems also tend to be financially expensive. This is indeed the case with Golly. Since our system required two workstation, an LCD projector, and a camera, the cost would exceed a more traditional system. It could also be economically prohibitive to projector / camera systems on a large scale. However, our system was built on commodity hardware and that it is likely that LCD projectors and cameras will soon be cheaply available. It seems safe to presume that at some point systems such as the one described here will also be inexpensive. Indeed, 'lighttables' have become a common fixture in many research labs. In the interim, the relatively low-cost projectors available today can be made to serve a large number of mundane surfaces with active mirror systems[10].

3 Implementation

The system we implemented is governed by the design philosophies of parsimony and transparency articulated above. The system itself consists of a light-table comprised of a video camera and projector situated above a *go* board on normal table. The system projects visual annotations that form a superset of the traditional board functionality: a game clock, a remote or artificial opponent's moves, and a minimal interface for exploring game variations. The vision system explicitly supports our design philosophy by accommodating the traditional style of game play. Unlike some other tangible user interface (TUI) light tables, like Underkoffer and Ishii's URP[14], our system relies solely on unmodified, traditional artifacts. Moreover the system is adaptive to various lighting and geometrical arrangements of the *go* board. What this means is that users interact with the light table-enhanced *go* board as they traditionally would, but are also provided useful augmentations. Consequently, players already familiar with *go* can learn to use the interface quickly because we've minimized the behavioral adjustments needed to use our interface.

3.1 Virtual Go Clients

GOLLY replaces the software-only *go* client. Software-only *go* clients all faithfully recreate the traditional *go* board inside the computer: they all allow the user to play a game of *go* on the computer. In addition to this basic functionality, all programs also add game mechanics augmentations such as automatically removing captured stones, automatically scoring games, navigating the game history, and saving and loading games in a standard format. Some clients allow annotating games, creating game variations (branches in the game history), playing games against other people via a number of Internet *go* servers, or playing against an artificial intelligence.

3.2 GOLLY Interaction Examples

To begin interacting with our system, a user can take out any unmodified *go* board, and place it on a table. Our vision system locates the board, and then applies a projective transform to recover the board's image. This means that the user does not have to concern themselves with calibrating the system. They simply interact with the system as they normally would: taking out the *go* board and starting a game.

3.2.1 A Regular Game

In a regular game of go, players alternate placing white and black stones on the board. Captured stones are removed, but otherwise once the stones are placed, they stay where they are until the end of the game.

Stones placed on the augmented board are seen by the system. The order of appearance of stones and the current state of the board are explained by a hypothesized sequence of valid moves. This sequence of moves is saved in the standard Smart Game Format (SGF)[5]. The process of explaining the board also involves identifying captured stones in the internal board representation. Discrepancies between the current observed board state, and the internal explanation are displayed onto the board via a projector: stones that should not be on the board (stones that have been captured but not removed, for example) are marked with a red \times , pictures of stones are projected onto the board to represent stones that are missing. In addition, the last move is marked with a red box. these augmentations can be seen in the figures above.

If two people play a game on the board, then apart from the current move indicator, the players may never notice the augmentation. The end result will be a standard game of go with the exception that there will be an electronic history of the game in the output SGF file.

Electronic records of games are very useful since it gives players the opportunity to go back through a game and re-examine decisions they made in the context of the completed game. On the other hand, playing a game of go on a real board opposite a real opponent is an important social interaction. GOLLY provides the opportunity to retain the physical and social environment of the real go board, while allowing the players to have an electronic record of their game.

3.2.2 Exploring Variations

students of go commonly explore game variations. A variation is a branch in the otherwise linear sequence of moves: a possibility to be explored. An exploration of possibilities happens every time a player makes a move, and studying variations on a physical board is an important exercise for improving one's ability to visualize moves. The subject of study may be a sequence of moves in the middle of a current game, or an isolated exercise from a book or newspaper. Unfortunately it is difficult to reliably return to previous board positions without help: typically a written record of the game or exercise. GOLLY provides a fluid mechanism for exploring variations on a real board with a minimum of visual clutter.

3.2.3 Studying Historical Games

Another common task is to study historical or previous games. By pre-loading the move history from an existing SGF file, it is possible to explore an historical game using the same variation mechanism described above. A visual annotation of the next move in the move history makes the process of moving forward in a game as fluid as navigating a variation. The same mechanism can be used to explore exercises: instead of transcribing a daily exercise from a newspaper, the exercise could be loaded into the GOLLY move history from a database on the Internet.

3.2.4 Internet and A.I. Play

Playing games over the Internet is a very popular activity, but usually, for obvious reasons, requires the use of go software. GOLLY provides a way to play remote games on a real go board.

By accepting moves from a module that interacts with the Internet Go Server it is possible to play games with remote users. Stones placed by the remote user are projected on the board. Stones that are removed by the remote user after being captured are marked with a red \times to make it immediately apparent what the local user must do to bring the physical board back into a valid state.

Playing go against artificial intelligence players (AIs) is a task that GOLLY can easily support. There are several AIs that play on the Internet Go Server (IGS). It would also be possible to attach a local AI to GOLLY using exactly the same move negotiation mechanism that is used to relay moves between GOLLY and IGS.

3.3 Limitations of GOLLY

The last two interaction examples push the limits of GOLLY. Selecting a historical game to study is usually done by finding the game in a book or in an online database. Playing a remote player involves finding a player and negotiating particulars such as board size, and move clock time. These activities present a difficult challenge for GOLLY if we strictly adhere to the tangible intermediary stricture and resist the temptation to project GUI elements like pop-up menus, Internet browsers, or chat client onto the board.

The current implementation requires the user to perform abstract interactions on an abstract interface before interacting with the physical board. The best way to solve this shortfall in the end might be to blend tangible intermediaries with other ubiquitous interface design components. These more abstract interactions (searching a database of games or finding a suitable opponent online) break out of the physicality of the go board, and are probably better implemented with another interactions modality, for instance, speech.

4 Experimental Validation

To test our assertions about the learnability benefits of tangible intermediaries, we conducted a study comparing GOLLY to a software-only go client. Time-on-task measurements and qualitative measures of participant success both indicate that GOLLY was easier for the participants to learn.

4.1 Experimental Design

In the study, we asked each participant to perform two similar tasks. These tasks approximated the act of studying an exercise on a go board. For each participant, one task was performed using GOLLY and the other was performed using the software client. We chose participants who did not know how to play go. The participants were told nothing about either system before the study.

4.1.1 Apparatus

The apparatus of the study included two instruction booklets, GOLLY, the software go client called GoStudio, a timing apparatus, and an exit questionnaire.

The tasks were specified in a small booklet of instructions. A booklet was comprised of several sections: a front page with initial instructions, several pages specifying the first task, a page asking them to notify the proctor and switch to the second system, several pages specifying the second task, and a final page again asking them to contact the proctor. There were two booklets: one specifying that the user was to start by using GoStudio and one that specified that the user was to start with “an augmented go board.” The two books were otherwise identical.

The first page contained the following statement:

The goal of this experiment is for you to use two systems to record games of the board game go. Please follow the directions on each of the following pages. Complete one step before looking ahead at the next pages.

This statement was followed by a statement asking them to sit down in front of the first system: either the computer console with GoStudio, or the augmented go board.

The tasks themselves were adaptations of problems from a popular beginner’s book on go[7]. The specific problems were chosen because they were approximately the same complexity, had natural variations that might be explored by a student of go, and did not involve any captures. The instructions did not include a primer on go, and the pilot study indicated that capture events were confusing. The tasks are shown in Figure 6 in the time-compressed format that is standard for printed works on go. In the instruction booklets the variations were unrolled so there was only one move per figure, and only one figure per page.

Each task consisted of approximately twenty pages. The first page instructed the participant to “Add pieces until the game looks like this.” and bore a figure similar to that shown on the left of Figure 6. The next ten pages each instructed, “Now add one piece.” and bore a figure with a single additional stone that wasn’t in the previous figure. The new stone was marked with a number and was the only stone so marked on that page. At the end of these pages the participant should have re-created the situation shown in the center of Figure 6 on whatever interface they were using. The next page then asked the participant to, “Remove pieces until the board is returned to the situation in Diagram 1, before you added the first piece.” and bore a figure identical to the first page of the task as a guide. This page was followed by a sequence of single move pages leading the participant through the second variation, ending at the situation shown on the right side of Figure 6.

GOLLY was used exactly as described in Section 3. GOLLY was not pre-loaded with any SGF files, so there were no annotations on the board until stones were placed by the participant.

The software go client GoStudio was chosen for the study after evaluating several free clients. GoStudio was chosen because it mostly closely followed the traditional Windows application paradigm, and therefore presented the best opportunity for generic Windows skill transfer. Specifically, GoStudio supports roll-over tool tips, an icon-based tool bar (similar to the tool bar in Word and Netscape Communicator), and the Windows online help system. GoStudio also had the easiest interface for creating variations. We were surprised to find that many clients make the act of creating a variation rather difficult. In GoStudio, if you use the history tool to go back in a game, simply placing a new stone starts a variation. Since we were

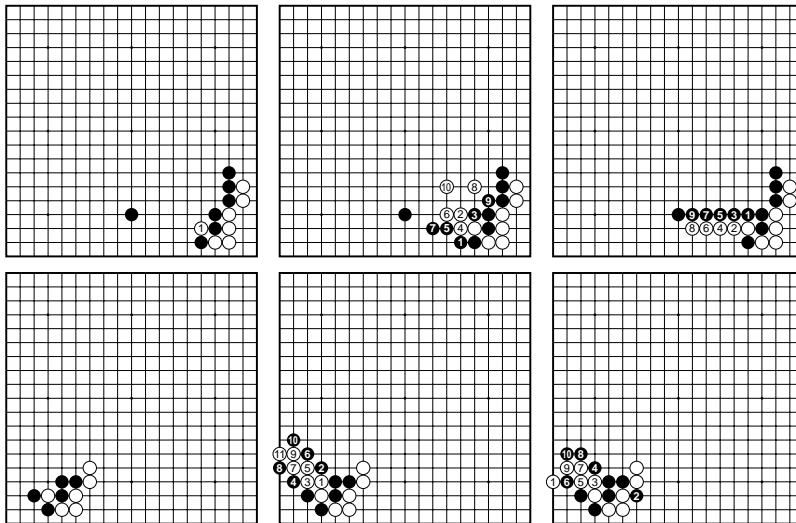


Figure 6: Two exercises, top and bottom, from Kaoru Iwamoto’s go book[7]. From left to right: initial setup, first variation, second variation.

asking the participants to create a variation. and GOLLY does this automatically, it was important to chose a software client with similarly fluid variation handling.

The proctor took notes on the interaction in Emacs. Each note was preceded by calling a command that inserts the current data and time into the buffer. Task times were then computed from these timestamps.

Finally, the participants were given an exit questionnaire.

4.1.2 Methodology

Participants were recruited by postering areas along the subway line near our lab. Posters enticed potential participants to “Earn money by playing a game” and directed them to a URL. The tear-off tag contained only a URL to our automated registration page. This pre-selected participants with at least modest computer skills, since they needed to access a web page to participate. It was decided that it wasn’t necessary to pre-select for participants without go knowledge, since the poster didn’t mention go specifically and we didn’t expect *a priori* to be inundated with go players.

Upon visiting the web page, participants were asked for contact information and to select a time slot from a list of open slots. Participants received a confirmation email with the date and time they selected, and directions to the lab. A reminder email was generated the night before the study. None of these web pages or communications contained any information about he study except to refer to it as “the Game Board Study.” The automated system randomly assigned the participants to one of two conditions.

The two conditions correspond to the independent variable: the system used by the participant to complete the task. Each participant worked through two tasks, one on each system, but the order that the systems were used was determined by the randomly assigned condition.

Subject	GOLLY	GoStudio
0-1	4.58	7.97
0-2	7.62	10.13
0-4	5.27	8.65
0-6	4.20	11.45 †
0-7	4.77	10.08
1-0	3.78	11.45 †
1-1	4.35	5.83
1-2	4.93	11.45 †
1-5	4.28	11.45 †
1-6	4.28	11.45 †
mean	4.806	9.991
s.e.	± 0.340	± 0.613

Figure 7: Time on task data, in decimal minutes. †denotes that participant Failed to complete the task before the time limit. s.e. is the standard error for the mean which gives the 95% confidence interval for the data.

Upon arrival the participants were asked to fill out payment paperwork. After that they were read an introductory script. They were handed an instruction booklet according to the condition number randomly assigned by the registration system, and directed toward GOLLY or the computer with GoStudio as appropriate to the condition.

During the experiment, the proctor took time-stamped notes indicating key milestones during the experiment including the moment that the participant began to read the first page of a task, and when the participant notified the proctor that they were done with a task. The time between the first page and the notification was later computed and taken as the time-on-task. If a task exceeded a time limit, the participant was stopped and moved on to the second task or to the exit questionnaire, as appropriate.

When the participant completed the second task, they were asked to fill out the exit questionnaire. After the questionnaire they were given an opportunity to ask questions. Payment was by check, sent the next week via mail to the address supplied on the registration page.

4.2 Results

There were ten subjects, five male and five female, between the ages of 20 and 41, with a mean age of 29.2. All participants described themselves as having “intermediate” computer experience, except one who chose “expert”. Five participants held Master’s or higher degrees, four held Bachelor’s degrees and one held a high school diploma.

The time-on-task data is shown in Table 7. Subject designations are split into a condition number followed by an ordinal. The subjects in condition zero used the GOLLY system first. The subjects in condition one used the GoStudio software first. Times of 11.45 minutes indicate sessions that were truncated by the proctor. Subject designations were assigned by the registration system, so missing ordinals indicate participants that didn’t keep their appointments.

Subject	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0-1					1	1	1
0-2	1	2	3	1	1	1	1
0-4	1	1	1	5	1	1	2
0-6	3	5	2	1	1	1	1
0-7	2	3	3	2	1	1	1
1-0	1	4	4	1	1	1	1
1-1	1	3	4	2	1	1	1
1-2					1	1	1
1-5	1	3	3	2	3	1	2
1-6	1	4	4	1	1	1	1
mean	1.375	3.125	3.000	1.875	1.200	1.0000	1.200
s.e.	± 0.246	± 0.412	± 0.356	± 0.449	± 0.190	± 0.000	± 0.126

Figure 8: Participant answers on the questionnaire. s.e. is the standard error for the mean which gives the 95% confidence interval for the data. See the text for details.

The questionnaire data is also shown in Table 8. The participants were asked to answer, on a five-point scale:

1. How hard was GOLLY to use? (1=Very Easy, 5=Very Hard)
2. How hard was GoStudio to use? (1=Very Easy, 5=Very Hard)
3. How useful were the visual annotations on the augmented go board?
(1=Helpful, 5=Confusing)
4. How would you rate the two systems for the task?
(1=Prefer GOLLY, 5=Prefer GoStudio)
5. Prior to today, how much did you know about the game go?
(1=Nothing, 5=Have Played)
6. How often, if ever, do you play go on the computer? (1=Never, 5=Daily)
7. How often, if ever, do you play chess on the computer? (1=Never, 5=Daily)

The first two participants (0-1 and 1-2) were given an abbreviated questionnaire that only contained the last three questions.

4.3 Analysis

The mean time-on-task using GOLLY was 4.806 minutes, with a variance of 1.154. The mean time-on-task for GoStudio was 9.991 minutes with a variance of 3.763. Half of the GoStudio participants reached a time limit, and did not finish the task. Allowing these participants to work to completion would have resulted in a higher

mean in the GoStudio case. Given the null hypothesis that the mean times are the same, but with unequal variances, the two-tailed probability that this data represents the null hypothesis is $\mathbf{p} = 3.3817 \times 10^{-6}$.

In all cases, the participants were faster using GOLLY than using GoStudio. Given the null hypothesis that they should be faster on GOLLY only half the time, the binomial probability of all trials being faster is $\frac{1}{2^{10}}$, or $\mathbf{p} = 9.766 \times 10^{-4}$. The mean answers to the questionnaire are also shown in Table 7.

Interpretation The quantitative data shows, with very high confidence, that participants were faster using GOLLY than GoStudio. Furthermore, the qualitative data from the questionnaire shows that the participants felt that GOLLY was easier to use, and overwhelmingly preferred GOLLY.

5 Contributions

This work focuses on a design principle for augmenting the traditional, physical tasks that consume a large part of everyday life. This is in contrast with much of the tangible interface literature that focuses instead on the useful, but different task of giving graspable manifestation to digital information [12].

This focus on existing artifacts demands transparency and parsimony. Transparency seeks to minimize the impact on the constellation of existing human behavior that surrounds the host artifact. Parsimony supports transparency by minimizing the clutter that distracts attention, and avoiding inappropriate metaphors that add cognitive load. Parsimony is also important for preserving the aesthetics of the host artifact. This minimization, avoidance of complexity, and preservations of traditional aesthetics all contribute to the increased learnability of our interface.

The go board is a particularly challenging artifact because of its highly refined aesthetic and behavioral constellation. We have presented an augmented go board that successfully showcases a minimalist ubiquitous computing design approach based on parsimony and transparency.

A Go: A Brief Primer

Go is a game where two players take turns placing markers representing their side on grid points. The pieces are black or white stones which are placed upon a wooden playing board. While board sizes vary, beginners often start with a 9x9 and slowly work towards using the standard 19x19 grid. This offers the players 361 points that they fight to control.

The goal of the game is to surround as much territory as possible. This is usually achieved as the result of tactical skirmishes through which players attempt to surround and isolate groups of stones. A free piece of territory is one on which neither player has placed a stone. Free territory adjacent to placed stones is often referred to as a liberty. A piece (or groups of stones) is taken if all of its liberties are exhausted. The white stones in Figure 9 are in atari, they have one liberty left and are in danger of being captured.

The game ends when the two players both pass, or one of the players resigns. The board position in Figure 10 shows a game at completion. A player's score is the number of free points with the addition of the number of his opponent's stones taken and the dead stones remaining in captured territory. If there is a disagreement about whether a group is dead, the players resume play until certainty is reached. The game provides

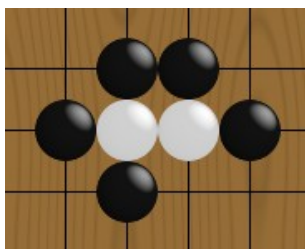


Figure 9: A white group in atari, the two white stones together only have a single liberty.

full-information to both players, and is a game of pure skill, as it involves no chance.

Go is said to mirror a tumultuous land battle in which several pockets of fighting are jumped between as players jockey for initiative and advantage. As the game progresses, groups become entrenched, and gradually all of the contested territory is divided between the opponents.

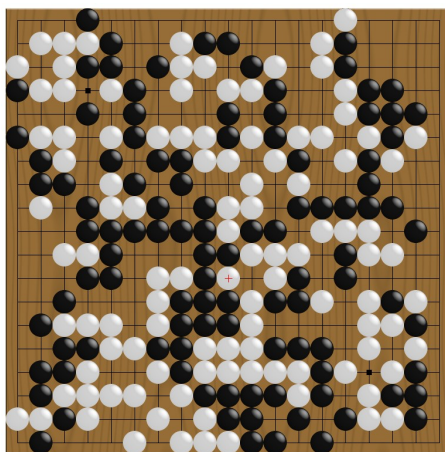


Figure 10: A game played in 1831, at completion..

The study of go is a serious pursuit. Many professional players devote their lives to the study of go. Many thousands of historical games exist and are the subject of careful study and annotation. Serious players sometimes memorize entire games as objects for contemplation. Different patterns of stones are endlessly debated and sometimes presented as problems to be solved.

References

- [1] Aaron Bobick, Stephen Intille, Jim Davis, Freedom Baird, Claudio Pinhanez, Lee Campbell, Yuri Ivanov, Arjan Schutte, and Andy Wilson. The kidsroom: A perceptually-based interactive and immersive story environment. *Presence*, 8(4):367–391, 1999.

- [2] P Boehner. *Ockham: Philosophical Writings*. Thomas Nelson and Sons, 1957.
- [3] J M Carroll. *The Nurnberg Funnel*. MIT Press, 1990.
- [4] Robert Headon and Rupert Curwen. Ubiquitous game control. In *UBICOMP Workshop on Designing Ubiquitous Computing Games*, September 2001.
- [5] Arno Hollosi. Sgf file format, 1999. <http://www.red-bean.com/sgf>.
- [6] H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of Conference on Human Factors in Computing Systems (CHI 97)*, pages 234–241, Atlanta, Georgia, 1997. ACM Press.
- [7] Kaoru Iwamoto. *Go for Beginners*. Pantheon Books, 1977.
- [8] M. Johnson, A. Wilson, B. Blumberg, C. Kline, and A. Bobick. Sympathetic interfaces: using a plush toy to direct synthetic characters. In *Proceedings of CHI*. ACM Press, 1999.
- [9] James Patten and Hiroshi Ishii. A comparison of spatial organization strategies in graphical and tangible user interfaces. In *Proceedings of Designing Augmented Reality Environments (DARE '00)*, pages 41–50, April 2000.
- [10] Claudio pinhanez. The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *Proc. of Ubiquitous Computing*, Atlanta, Georgia, September 2001.
- [11] Ben Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Intelligent User Interfaces*, pages 33–39, 1997.
- [12] B. Ullmer and H. Ishii. Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, 39(3&4):915–931, 2000.
- [13] B. Ullmer and H. Ishii. *Human-Computer Interaction in the New Millennium*, chapter Emerging Frameworks for Tangible User Interfaces, pages 579–601. Addison-Wesley, 2001.
- [14] J. Underkoffler and H. Ishii. Urp: A luminous-tangible workbench for urban planning and design. In *CHI*, pages 386–393, 1999.
- [15] P. Wellner. The digitaldesk calculator: Tangible manipulation on a desk top display. In *Proc. ACM SIGGRAPH Symposium on User Interface Software and Technology*, pages 107–115., 1991.
- [16] C. Wren, F. Sparacino, A. Azarbayejani, T. Darrell, T. Starner, Kotani A, C. Chao, M. Hlavac, K. Russell, and Pentland A. Perceptive spaces for performance and entertainment: Untethered interaction using computer vision and audition. *Applied Artificial Intelligence*, 11(4):267–284, June 1997.
- [17] Christopher R. Wren and Carson J. Reynolds. Minimalism in ubiquitous interface design. *Personal and Ubiquitous Computing*, 8(5):370–373, September 2004.