

The Computer's Role in Sketch Design: A Transparent Sketching Medium

Michael Trinder

*The Martin Centre for Architectural and Urban Studies,
University of Cambridge Department of Architecture*

Key words: sketching, sketch design, user-interface, transparency, immersion, computer-aided design

Abstract: Starting from an analysis of the current unsuitability of computers for sketching, three key requirements are identified, in particular the notion that re-drawing or over-drawing are more important than editing and tweaking. These requirements are encapsulated in the broad concept of Transparency, understood both literally and metaphorically. Two experiments in implementing aspects of Transparency are described. One subverts the Macintosh window manager to provide windows with variable transparency, so that tracing between applications becomes a practical possibility. The other implements a graphical interface that requires no on-screen palettes or sliders to control it, allowing uninterrupted concentration on the design in hand. User tests show that the tool can be learnt quickly, is engaging to use, and most importantly, has character.

1. INTRODUCTION

Computers are being used by architects almost exclusively for the later stages of design: one UK practice has been quoted as using CAAD for upwards of 90% of their issued drawings while early presentation work and sketches are made with pencil and paper. Under the temporal, financial and aesthetic pressures of the design office any tool will be used only for tasks that designers perceive are within its capability. Whether their current perception is based in prejudice, preference or past experience, designers avoid using computers for sketching. The software currently available for sketch design is plainly not working.

Of course, the lack of useful software is not a hindrance to design. Buildings have been and are built without any need for information technology; design is

famously as at home on an envelope, a cigarette packet or a cocktail napkin as it is on the drawing board or MacPaint. However, architects using CAAD to produce hard line drawings, but only pencil and paper to sketch, experience real problems when moving between these contexts. Electronic drawings must be printed out before they can be reworked. Alterations must then be copied tediously back into the computer. If the computer could support sketch design, the architect could work in either context, as they see fit.

This paper presents a small part of a PhD investigation into how to design interfaces that support sketch design. Our approach to the problem is to understand that sketch design and the computer as having some fundamental differences that need to be recognised before they can be overcome. We see the computer's role in sketch design not as some design companion but as presenting a fluid and engrossing sketching medium, stage managing the transition of drawings from the back-of-the-envelope to hard line drawings, renderings and models and back again.

2. RELATED WORK

Existing work that is relevant to the transparent medium and sketching tool presented here can be divided into two broad categories: projects concerning issues of transparency and those investigating sketching. The amount of work dedicated to the latter far outweighs the former, and much sketching/painting research has little relevance to the proposals for a transparent interface. However, a few projects which tackle the sketching discipline have used aspects of transparency, and it is on these that we concentrate.

2.1 Sketch design software

The Cocktail Napkin [Gross 1994, 1996] concentrates on gesture recognition to interpret sketches into structured diagrams. The marks made cannot be separated or recombined in new ways, nor do they lend themselves to re-interpretation or ambiguity. PerSketch [Saund, Moran 1994] attempts to rectify this shortcoming by giving the user 'access to visual objects as they are perceived by the human visual system.' This is achieved by retracing existing objects to work with emergent forms.

Neither the Napkin nor PerSketch fit our approach to sketching. Piranesi [Schofield 1994], [Schofield, Richens 1995] comes closer in an attempt to extract the rendering process from the parametric, trial and error realm and allow selective, interactive rendering by using geometry derived from a 3D image's Z buffer.

Finally, a realised building project, the Groningen Museum by Coop Himmelblau makes an interesting example of what architects *expect* from sketch design software. [Rattenbury 1995] describes the process: 'an original cryptogram, overlaid three times, was converted into a model and then scanned straight into a computer using sophisticated software and transformed into plans "since we didn't care too much about the functions of the space." The computer in Coop Himmelblau's office was then linked to a computer in Groningen's redundant dockyards where the gallery's huge steel panel components were welded together by the shipbuilders and then transported to the site for assembly.'

Coop Himmelblau may be interested in an automatic building solely for rhetorical reasons, but they see computers as being able to cope with the shifts of

interpretation necessary to design buildings through drawings. Most of all this demonstrates that it is possible, with some effort, to produce architecture within the digital medium.

2.2 Uses of Transparency

[Colby & Scholl 1991] provide some important groundwork, showing how transparency and blur can be used to prioritise information within a complex display and relating scales of each to visual impact. [Staples 1993] examines the 'desktop' GUI as an artistic artifact, drawing analogies between the use of perspective and transparency in painting and their application to interface design, noting that the latter confers supernatural and special status on objects in addition to allowing multiple images to overlay. Although this is a purely theoretical paper, it does contain a mocked up image of a transparent window which comes closest to the medium we describe here.

PAD [Perlin, Fox 1993] and The Macroscopic [Lieberman 1994] use transparency to allow 'semantic zooming'. This involves slowly replacing abstract information with more detailed versions as the user zooms in closer.

[Kamba *et al* 1996] use transparency to allow multiple interface widgets to occupy the same screen space so as to free up the small displays of most hand-held PDA devices. Unfortunately for their experiment, they imagine their widgets as being above the information on display, yet render them as if they were below it. What they show is that transparency does not work when the user's perception of depth differs from that maintained by the interface. [Harrison *et al* 1995a, 1995b] and [Harrison, Vicente 1996] examine the user of transparent interface widgets in extensive detail in order to determine the optimum levels of transparency for these elements. Their intention is to prevent dialog boxes, menus and icons from taking up valuable screen space in graphic intensive applications such as 3D design rather than to exploit overlaid images.

[Bier *et al* 1993,1994] present a compelling use for such transparent interface widgets. Their Toolglass has been the foundation for a large number of papers which form the most extended investigation into Transparency undertaken so far, passing images up and user interaction down through a stack of toolglasses to allow the user to construct an interface out of the overlapping elements. Importantly, the documents that the user is working on are always shown as opaque. The transparent medium that we present here makes *every* window transparent, removing the distinction between interface tools and documents.

[Carter 1993] and [Wellner 1993] employ a combined overhead projection system and camera to set up a visual feedback loop similar to [Kreuger 1991]'s installation work. Their DigitalDesk system overlays a digital desktop onto a real one, allowing hand drawn images to be instantly digitised and copied or used as the basis of digital transformations. DigitalDesk is the most literal, and successful, expression of the transparent computer.

Finally, [Kramer 1994, 1996] implements a sketching environment from a stack of freeform transparent sheets of drawings and, as such, provides the closest parallel to our work on a transparent medium.

3. SKETCHING AND SKETCH DESIGN

3.1 Sketching Theories

Studying sketching is fraught with potential confusions. The biggest problem is linguistic - the words *sketch* and *drawing* refer both to the act as well as its end product, the act of sketching is seen to be different from the image that it produces. To help distinguish these, *sketch* will be used to refer to drawings that are being made and *image* to the final result. The image is made up of *marks* and sketching can be seen as the act of *mark making*. Quite why these distinctions are important to sketch-design software becomes clear after digging a little deeper into the nature of sketching.

Today design is inexorably entwined with some sort of drawing. Whether this is because drawing is a natural design medium, a historical accident or even a device to exclude the layman is not really an issue. What is more important is that there has been a tendency for theories of sketch design to take this relationship as read and concentrate solely on sketching as an act - looking at the marks made - and the images produced. Whilst the *interrelationship* between sketching and design is not in question, failing to assess this relationship leads to serious confusions in understanding just what is so important about sketching.

It is usually understood that the image *is* the design: perspectives, for example, are presented to both clients and planners as objective tests of a design's validity. [Pérez-Gómez 1983] shows how this relationship between architectural drawing and design developed, in the mid seventeenth century, from a functional view of reality. The proposal that the world can be represented in an objective, self consistent manner developed at this time underlies our contemporary appreciation of visual truth: images are to be judged by how well they describe the world. [Coyne 1995] outlines the two most prevalent explanations of how images are able to encode reality, the Correspondence and the Constructive theories, before going on to propose a theory based in Phenomenology.

Phenomenology sees everyday experience and action as activities of complete immersion, of being thoroughly engaged in doing. The intellectual stances of subject and object represent a breakdown of this engagement. Applying this theory to architectural representation, [Veseley 1993] concludes that it is meaningless to search for information in sketches as the images only reveal design rather than encompass it. Architects can be surprised by their sketches and find new ideas in them, but this is not because the idea is actually *in* the image. Rather, the act of drawing helps to reveal the idea, revelation that is not possible *without* the image but entirely separate from it. They are reacting less to what they see than what they have thought about whilst doing.

3.2 Computer-based sketching

Drawing is a pre-reflective, immersive state that is culturally based and culturally biased. As [Veseley 1993] explains, any drawing act is made as part of the established cultural context that allows us to create, understand, share and experience space without being able to represent it in its entirety. However, if the sketching experience is one of undifferentiated involvement, then any intervention into this, occasions when *things* suddenly intrude, times of breakdown, will halt the

drawing prematurely. The decision to treat a drawing objectively must only be that of the sketcher and not one imposed by a data structure unable to support drawing or an interface that gets in the way. Sketching depends on lack of constraint and on fluidity.

If the phenomenological viewpoint is correct, the computer must be interfering with the pre-reflective immersive state that the sketching act requires. Although the computer tries to present an electronic version of your desktop, this desktop is subtly different in a number of important ways:

- Every action is limited in some way. There are very few fundamental limits, things that the computer cannot do at all, but every application imposes restrictions on what the user can accomplish.
- A computer application is an encapsulation of its programmer's understanding of a task. It is not possible to rotate text in Microsoft Word, you cannot blur a drawing in a CAD system, you cannot draw a continuous diagonal line in Photoshop. Whilst object-oriented software and integrated 'office' packages attempt to get around the problem it is important to recognise that the limitations still exist.
- Whilst the modern GUI has helped users learn how to use applications more quickly by presenting their interfaces in the same way, this visual similarity unfortunately implies that the underlying applications function in the same way.
- Furthermore the interfaces get in the way, eating into the limited screen area available to the user. Even when most of the interface is placed into floating palettes a drawing can be almost completely obscured.
- Electronic documents are infinitely mutable – there is no record of a drawing's evolution beyond what is explicitly recorded by the computer or user. Whilst this is a definite advantage for drafting, drawings are no longer definitive records.
- Most CAD applications use a selection and manipulation tool (usually represented by an arrow) as the default tool. After drawing an object, the current tool automatically switches to selection. This promotes editing and manipulation of a drawing over creating new marks – the user is encouraged to switch from drawing to analysis of the drawing. Such applications naturally interrupt any immersive state and place emphasis on the product not the act.
- The tendency to slip into continuous editing can also seriously compromise the quality of work carried out electronically: an investigation by [Penz & Wright 1994] to find differences between paper and digital design shows that design students evaluated fewer alternatives at a basic level when computers were involved. This tendency, termed the 'Icon Trap' in [Lawson 1990], by which a single image or set of images come to *be* the design, blinds the designer to any investigation other than that dealt with in these drawings. [Jansson, Smith 1991] and [Purcell, Gero 1996] make specific studies of this 'design fixation' and show how images given to test subjects can shape the way they subsequently design, even reproducing faults inherent in the examples. The cure lies in first realising the iconic status of a drawing (usually an early concept statement) and setting it aside for new work to start. This is less easy to do in a digital medium that is seamlessly

mutable and pitched toward editing existing documents rather than re-creating them.

- Computer displays have an extremely low resolution when compared to that of paper and pencil.
- The mouse is a terrible drawing instrument, closer to a brick than a pencil. It was designed purely to give two dimensional positional information - for *pointing*. Its design restricts movement of the hand and arm to allow only this pointing movement. Its standard shape forces the hand and arm into a pointing gesture with forefingers extended to press buttons and the wrist locked in position, a straight line running from elbow to finger tip. It is at best extremely difficult to move a tool fluidly when in this position, let alone use of the fingers, wrist and elbow to vary the orientation and pressure with which that tool is used. [Zhai, Milgram, Buxton 1996] show that devices that take input only from the arm or hand instead of the fingers are inherently less flexible and accurate.
- Finally, modern graphic interfaces use opaque windows. Architects don't use opaque media very often – they use translucent and transparent media such as tracing paper and polyester film to copy, elaborate, annotate and evolve drawings. The only opaque drawings usually to be found in an architectural practice are mechanical reproductions.

3.3 The Computer's Role: Transparency

We propose that the concept of *transparency* has significant potential for the development of creative electronic media. The term transparency implies much more than a purely optical effect, extending to ideas of ubiquity, unobtrusiveness and interchangeability. It has been an ideal of the Modernist movement in architecture, only truly realisable with recent innovations in glass technology. Taking the broader meaning, *transparency* can be used to investigate four separate domains:

Computer Transparency

The degree to which the physical form of a computer's input and output devices promotes intuitive interaction.

Interface Transparency

The degree to which the user interface promotes the experience of immersion.

Data Transparency

The degree to which user actions are freed from the limitations imposed by a particular data structure.

Media Transparency

The degree to which conceptually separate groups of data can be combined visually.

4. A TRANSPARENT MEDIUM

With user experience of windows being based entirely on a stacking illusion, it would seem to be a trivial extension of the desktop metaphor to let any covered images retain their presence. Turning the standard Macintosh document window into a transparent medium introduces transparency within an easily understood context, adding new functionality to an existing interface element rather than re-inventing the entire interface.

4.1 Simulating Transparency

There are a wide variety of transparent and translucent media in use by architects although these are usually referred to by only three names: drafting film ('Film'), tracing paper ('Trace') and, more rarely, detail/layout paper. Superficially these papers all appear very different but, in use, significant characteristics become clear that are much more consistent. Transparent media are used not only to bring images together as one, but also to place images in some sort of hierarchical relationship: the partial transmission of the image diminishes its visual significance letting the upper drawing read as the new, revised or more important version. Degrading images is important to the sketching process, letting earlier images fade in order to create new ones, as is the ability to distinguish between drawings so that, when simply copying from one sheet to another, it is possible to determine what is still left to transfer. These media degrade and separate images by reducing their contrast, altering their colour balance and by subtly blurring them.

Creating a window that has similar properties turns out to be more complex than it would at first seem. All computer simulations of transparency are generated by some sort of blending function that produces a single output, the colour to be displayed, from a pair of inputs, the images to be overlaid. At its simplest level, pixels are taken directly from one or other of the images according to a binary mask. This is exactly the technique adopted by [Perlin, Fox 1993] for their PAD interface. In this way images with non rectangular boundaries or containing 'holes' may be displayed by only transferring the relevant pixels. Extending this mask from a single bit per pixel to, generally, eight bits per pixel allows a mixture of colours within a pixel to be expressed. This extra information, the alpha channel, is usually conceptualised as the opacity of the upper or foreground image.

[Colby and Scholl 1991] stated that there is little perceptual difference between a physical transparency and a rendering in which one image can be seen to continue through another, in which the two images are blended by linear interpolation. There are however situations in which the results of alpha compositing are wholly different from that experienced using physical transparent media. The standard linear blend function is symmetrical; the designation of foreground and background is only meaningful in that the value α is defined in terms of the opacity of the 'foreground' image. To deliberately associate this value with a particular image, the notation $f\alpha$ will be used to represent the alpha value associated with the foreground image **F**. This foreground image will always be the uppermost of the two.

The standard linear blend formula shows how the output pixels are actually calculated:

$$\mathbf{I} = \mathbf{F}.f\alpha + \mathbf{B}(1 - f\alpha)$$

where \mathbf{F} = Foreground pixel, \mathbf{B} = Background pixel, $f\alpha$ = Foreground alpha value and \mathbf{I} = Resultant pixel for display

The problem with this standard blend is that the result of compositing two images with $f\alpha=1/4$ is identical to that of compositing them with $f\alpha=3/4$ with the notional stacking order is reversed (fig 1). Put another way, the stacking order we perceive may be different from that maintained by the computer. As [Metelli 1974] shows, we will read the stronger image as the foreground.

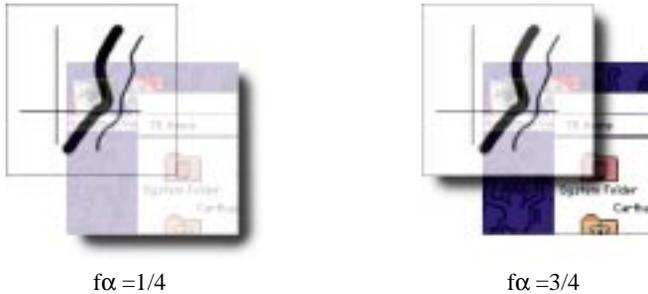


Figure 1.

$f\alpha = 1/4$

$f\alpha = 3/4$

Instead, we must use two successive blends. The first reproduces the effect of the medium, toning down and obscuring the images below based on one alpha value, $m\alpha$. Altering $m\alpha$ changes how much of the background image is visible behind the foreground, whose transparency remains constant. The second blend introduces the marks made upon the medium by blending the composite background image with the image the application wishes to display.

A series test blends were carried out using a typical piece of the Macintosh desktop and the foreground marks shown in fig 1. The first experiments simply mixed the background image with solid white (**W**) or a colour (**M**) according to $m\alpha$ using pattern dithers or alpha blending. Of the four experiments, the coloured alpha blended images were the most effective (fig 2), but all four suffered from the same flaw. At 100% transparency, when $m\alpha$ is zero, the background image is passed through untouched making it very difficult to distinguish. The perceived stacking order reverts to being based on the relative strength of the images. It is impossible to determine the actual stacking order of the images without trying to move one of them. Whilst the user may well wish to combine images perfectly in this manner at some stages in their working process, this tendency toward absolute transparency is an unfortunate one for a proposed sketch design environment - old drawings would continue to have the same impact as new ones.

The solution adopted was to reverse the behaviour of the medium's colour. Rather than have the medium gain colour as it becomes more opaque, it instead starts out tinted and slowly fades to an opaque white sheet. Fading the colour of the medium as transparency drops may not reflect any physical experience of transparent media but the result is that the opaque composite is indistinguishable from a standard opaque window whilst increasing transparency lets in a colour to help separate foreground and background (fig 3). As the sheet becomes more transparent its white surface fades away to reveal a tinted image beneath.

The final blend adopted went one stage further in degrading the underlying image by first passing it through a simple blur filter. Although only a minimal amount of blurring was used, it immediately separates the two images, the contents

of the window and the images that can be seen through it, into foreground and background. Thin marks in the foreground can be clearly distinguished as they float slightly in front of their backdrop.

In the final interface the medium's transparency is altered by a slider in the corner of each window. Moving the slider pops up a small preview frame to give real-time feedback of the current transparency setting (fig 4).

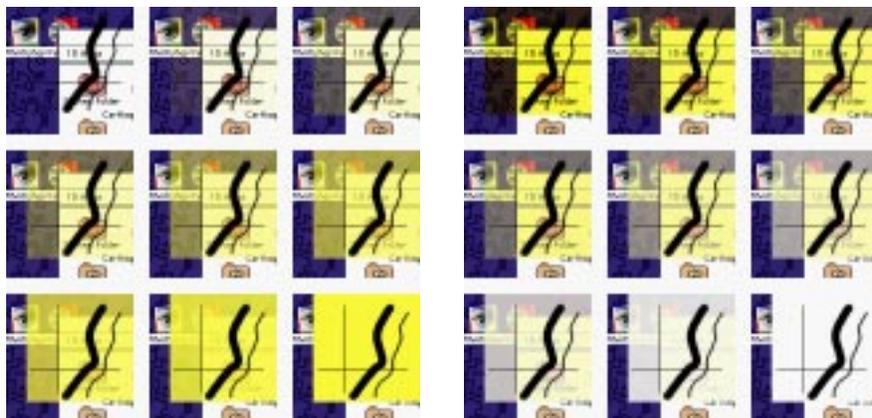


Figure 2 left. two successive alpha blends: $I = F.f\alpha + (1 - f\alpha).(M.m\alpha + (1 - m\alpha).B)$

Figure 3 right. Reversed colour blend with blur: $I = F.f\alpha + (1 - f\alpha).(W.m\alpha + (1 - m\alpha).f(B)).M$
 ($W = (1,1,1)$ RGB, M = medium colour, f) is a simple blur convolution kernel)



Figure 4. Altering a window's transparency in real time using a preview frame.

4.2 Using Transparent Windows

Even at their most basic level, simply overlaying one image onto another, transparent windows allow the user to work in novel ways that reflect the way physical drawings are used. An initial sketch can be overlaid, drawn again and again until the idea becomes clearer. Significantly, as more windows are overlaid, the original image is increasingly obscured as it fades, blurs and is covered by new marks.

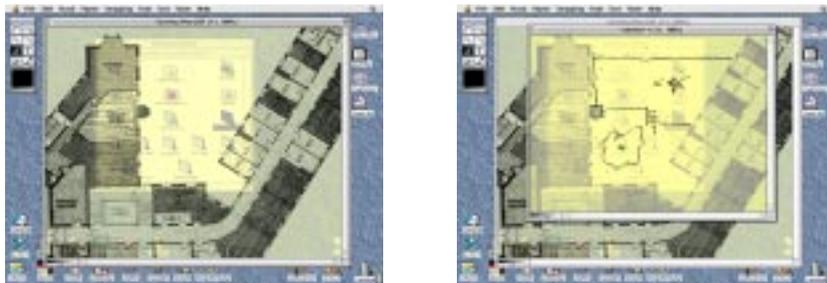


Figure 5 left. A CAAD plan with the Finder beyond
Figure 6 right. The plan overlaid by a new sketch of a proposed cafeteria

Combining the windows of different applications takes the idea much further (figs 5-9). A CAAD drawing can be sketched over to rework a detail, or a hard line drawing can be developed using a sketch as a guide. Text can be appropriated from a word processor (fig 9), images directly from scanning software or even a video monitor.

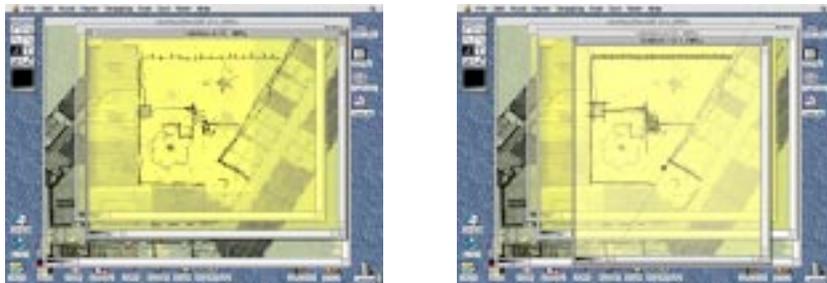


Figure 7 left. Another sketch with more detail overlays and starts to obscure the original
Figure 8 right. A two dimensional drafting window is added. Note the full screen snapping lines.

Simply overlaying windows in this fashion requires no alterations to existing applications as long as they do not draw directly to the screen. The transparent window exploits the fact that all applications must present their data as a pixel map in order to exist within a GUI. A custom Window Manager redirects application's display into an off-screen cache from which produces a composite. This leaves the

user to interpret the images as they see fit – something the human brain is extraordinarily accomplished at doing.

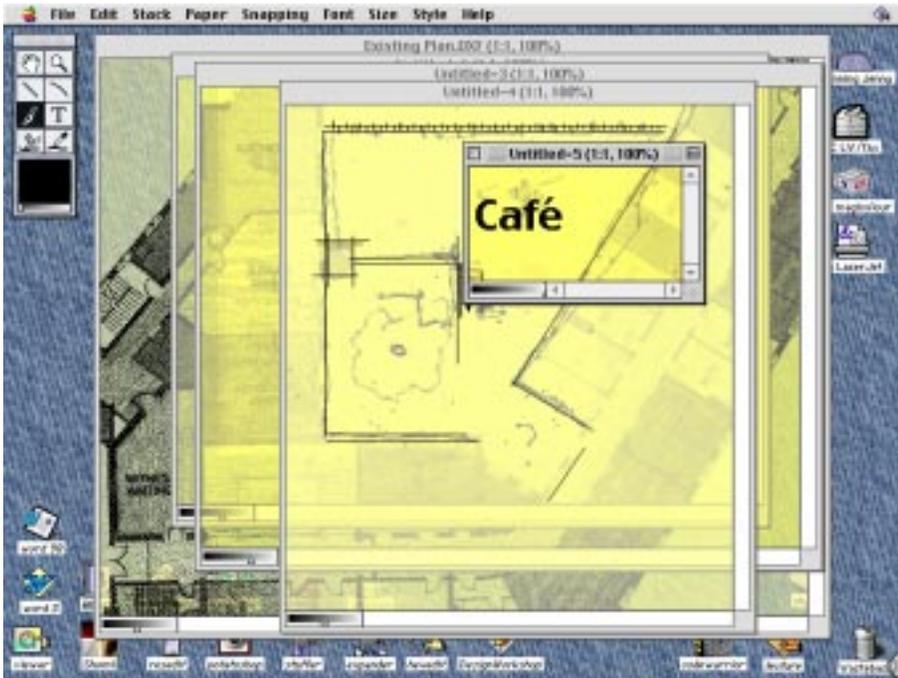


Figure 9. A text window is positioned to add an annotation.

With some simple additions to an application, the transparent medium can become much more than way of compositing images. As long as an application responds to requests from the window manager for the nearest geometrically significant point to a given co-ordinate, geometric snapping can take place between windows maintaining any data structure. Passing two-dimension geometry between windows is achieved by using the display as a standard co-ordinate system. Since snapping is no longer limited to a single window, construction lines must be drawn across the whole screen. Finding the intersection between vectors in two different windows within the co-ordinate space of a third is perfectly possible (fig 10).

Two special tools, designated *Push* and *Pull*, allow pixel map images to be pushed between windows that will accept pixel data. Most obviously, this allows the composite image produced by a stack of windows to be recorded by pulling the various images up into a new window that covers them all. A perspective rendering can be populated by opening stock images of people and rubbing over them with the push tool, as if they were dry transfers. Intermediate windows can be used as masks to limit the transfer to specific areas (fig 11). This is just not possible with opaque windows. The translucent windows allow the user to simultaneously see the source and destination images in alignment and carefully control the process.

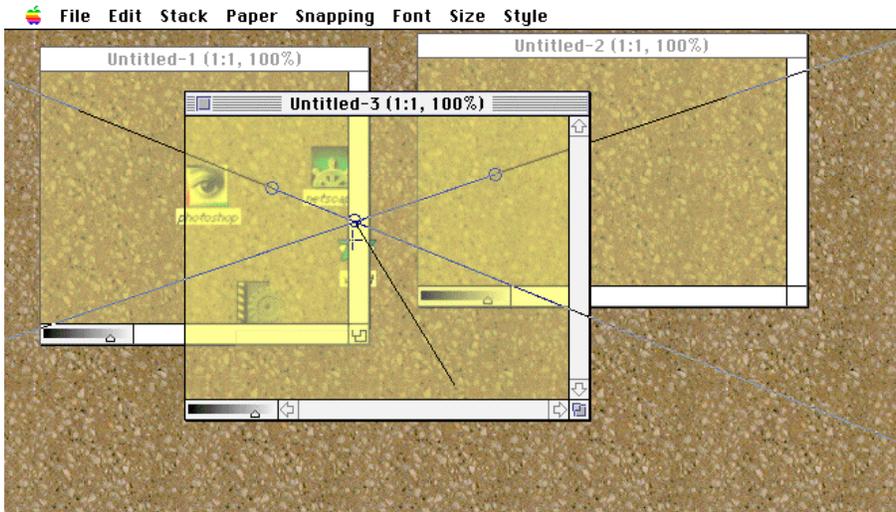


Figure 10. Snapping to the intersection of two discrete lines in a third co-ordinate space.



Figure 11. Transferring an image through a stencil: *Top left* The stencil in place over the receiving window. *Top right* The source image placed ready for transfer. *Bottom left* Transfer in progress. *Bottom right* The results in the lowest window

Finally, using a live video feed as a source gives the user a constantly changing palette of images and colours with which to paint, requiring skill and timing to use successfully. A novel creative medium can be built by the user by combining simple

applications whose hybrid functionality lies outside the scope of any of those applications. The example image (fig 14) was created from live television broadcasts, moving the video window to an appropriate position and then rubbing through the image before it disappeared. The unpredictable and uncontrollable nature of the source images mean that concentration is essential and that continuous drawing and redrawing is the only possible strategy.

5. A TRANSPARENT SKETCHING TOOL

Having already concluded that the mouse has, frankly, all the drawing ability of a brick, we chose a pressure-sensitive drawing tablet as the minimum level of hardware necessary for sketching. Not only does this afford a greater level of Computer Transparency by presenting the user with a familiar drawing instrument and a more natural hand position, the pressure-sensitive tablet offers an extremely rich stream of input. Table 1 shows how extensive this input can be.

Table 1. Stroke characteristics available with different input devices

Mouse	Tablet	Wacom UltraPad
X	X	X
Y	Y	Y
Button	Tip Pressure	Tip Side button Pressure Plan orientation Tilt Tool used
Velocity	Velocity	Velocity
Acceleration	Acceleration	Acceleration
Bearing	Bearing	Bearing
Curvature	Curvature	Curvature
	Rate of pressure change	Rate of pressure change

Table 2. The variety of marks made by the sketching tool: *left* using just pressure, *right* using pressure, velocity, bearing and acceleration.

High						
Medium						
Low Velocity						
	low	medium	high	low	medium	high
	pressure					

This sketching tool is a deliberate experiment in Interface Transparency. The tool is designed to be entirely controlled by the manner in which the stylus is used, making clean and ‘dirty’ lines, brush marks, hatches, area fills and splattery marks without any need for the user to select options or adjust sliders in a tool palette. The traditional palette of brush shapes and characters can be just as adjustable but the characteristics must always be adjusted before making a mark. The user must predict the marks they wish to make and use the stylus sometimes to draw, sometimes to select options. The sketching tool can be constantly adjusted and remain a drawing tool at all times.

The marking algorithm uses both a circular brush mark and a line drawn directly between sample points. The brush is scaled proportionally to pressure and inversely to velocity, so as to make it possible to hatch. Hatching involves repeated high velocity strokes using a thin line and an arcing motion of the hand – the velocity and hand motion combine to make fine pressure control more difficult. Reducing the brush size at speed compensates for this.

The brush is offset from the sample points according to acceleration: the acceleration value tends to be relatively constant during a stroke, but rapidly reaches extremes at the start and end of the stroke and at sharp changes in direction. Perturbing brush marks with acceleration lends rapid, sloppy strokes a noisier appearance than carefully controlled strokes and also produces blots and blotches at the ends of lines if the user stops dead instead of ‘following through’.

Connecting sampled points by a straight line not only copes with widely spaced samples but also has the useful effect of separating the mark into two parts: line and shading. At low pressures, when the added effect of velocity often scales brush sizes down to a single pixel, the tool generates curved lines that faithfully follow the stylus. As velocity increases, the sample points become more widely spaced, the level of approximation increases and the lines straighten out. Increased pressure in both cases thickens the line into a brush stroke and the tool becomes a shading rather than drawing device. In fact, at high pressures connecting sample points by a line detracts from the tool’s shading. Fast, high pressure strokes result in occasional brush marks strung along a thin line. To rectify this unfortunate situation, the line is not drawn above a 50% pressure level.

6. USER FEEDBACK

6.1 The Transparent Medium

The transparent medium has been demonstrated to two broad groups of people during its development: programmers involved in interface design and professional architects. Both groups suggested improvements, but there is an appreciable difference between the comments made by the two groups. When the software was demonstrated to those involved in programming and interface design, the general reaction was an appreciation of the visual effect and the implications this had for the interface, but few could see any use for it. It looked nice, but the main question was simply ‘Why?’ When design professionals were shown the same software their response was much more immediate: the overwhelming question was ‘When?’

This difference of opinion is not just a reflection of different attitudes to the computer. Of course a programmer would question the need for such a complex solution to an apparently trivial problem and of course an architect would empathise with something deliberately based on their working process. What is important is that designers could immediately see *how* to work with transparent windows. They recognised familiar approaches to design that their own computers would not support. Whilst this is reassuring and supports the theory that transparency will have positive benefits for sketch design, it also indicates that these techniques are not universally applicable. Some activities do not become any easier simply because the window is translucent. The computer programmer has little use for overlaid information or visual ambiguity. The large processing and storage overhead the translucent window entails is hard to justify in these circumstances.

6.2 The Sketching Tool

The proposed sketching tool has potential flaws. Its design was based on a theoretically strong but unproved assessment of the marks made during sketching. For it to be used with confidence and easily learnt, it should react in a controllable and predictable manner to the sort of movements made by the hand whilst sketching. If the user cannot control the marks made, does not discover how to make certain marks or creates marks that they had not intended then the sketching tool will not provide an immersive sketching experience. Its success depends on exactly the same criterion as the transparent medium – is it useful?

Twelve test subjects were taken from a cross section of the architectural profession, from first year students through to professionals with several years of post qualification experience. They were presented with a sketch executed with the prototype tool (fig 12) and asked to practice using the tool until they felt confident to make a copy of the sketch. Drawings were recorded as pixel map images as well as raw streams of time-stamped data from the tablet, allowing any session to be replayed in real time. Subjects' comments were recorded on tape.

Three measures were derived from the data – the time taken before the subject 'felt confident' with the tool (taken as being the time before an attempt was made to copy the sketch), the amount of control the subject had over the stylus (measured by looking for distinct groupings of pressure and velocity) and the fluency of the reproduction (how many types of mark they discovered). Of these, the time measurements show the clearest results (fig 13). The subjects fall into two surprisingly distinct groups when their contact times – the time spent with the stylus in contact with the tablet – are considered. Most went straight into copying the test image after only cursory experiments with the pen, whilst four spent nearly 2.5 minutes trying out different marks and lines. The faster group continued to make discoveries about the pen whilst drawing the final image: subject 4 for example only discovered the relation of speed to line weight right at the end of the test. The slower group produced a wide range of marks during their practice. The only obvious correlation between type of subject and the contact time is that all the students fall into the faster group.

Although this was an informal pilot study with too few subjects to allow statistical analysis, there are some interesting observations to be made from the way the subjects drew and the way they talked about these drawings. In fact recording their comments whilst drawing proves more useful than any of the analytic methods.

The overwhelming topic of conversation was the sensitivity of the pen to pressure, making marks too thick, and to acceleration, causing large splatters. The version of the tool used for the tests was deliberately extremely sensitive when used at low velocities - part of the experiment was to see if users would discover that line weight became more controllable with increased velocity. Unfortunately, because the test subjects were unfamiliar with the drawing tablet - a topic raised by four of them whilst drawing - and because they were copying the test drawing very carefully, they tended to draw at low speeds. When the user concentrates on getting one aspect of a mark correct - as these test subjects were - he or she is doing so to the detriment of the other variables. Furthermore, where a small amount of acceleration noise textured a line the subjects were pleased with it, but where there was too much splatter or where they couldn't get rid of the splatter they were less happy. Only a few discovered that drawing faster gave them more control over the lines. Whether this is because it is not intuitive or because of the care with which they were copying the sketch is not possible to say. Those that did work out the velocity control tended to say so out loud at the moment of discovery - 'Aha, that's what you want, it's thinner if it is quicker' or '...actually...the faster you move it, the lighter the lines you make with it'.

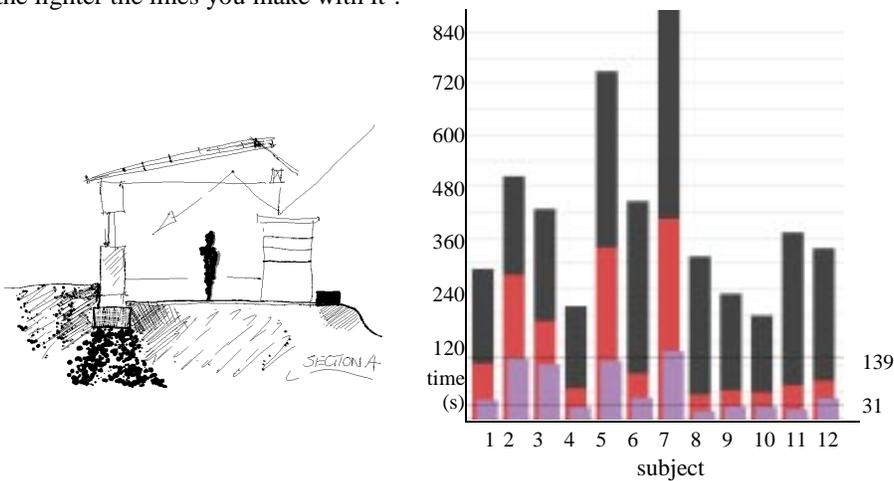


Figure 12 left. The sketch to be copied.

Figure 13 right. Total time (black), learning (grey) and actual contact times (offset).

Finally, some subjects had real problems orienting their strokes, coping with the dislocation between the stylus tip and the marks on screen and with the low resolution of their sketches. These are all problems of Computer Transparency that the sketching tool tested does not address.

The way subjects talked about their drawing proves to be almost as interesting as what they actually said. Whilst drawing, they would speak in incomplete sentences, only expressing fully formed ideas when they stopped to discuss something. These moments are excellent indicators of 'breakdown'. When subjects talked about the experience, there was a definite tendency to address their complaints to the stylus itself rather than the computer (or the monitor). There were also a number of comparisons between the pen and physical drawing tools - analogies to typewriters, Rotring pens, Sumi brushes, Croquil pens and Gerald Scarfe were made. They were

seeing the *stylus*, and not the software, as having these characteristics – it felt to them like a real tool with definite character.

That the subjects were able to produce engaging sketches whilst noting that the pen was sometimes ‘out of control’ indicates the demand that designers place on their drawing tools. Figure 15 shows the quality of image that some were able to produce. The test tool was extremely sensitive to acceleration, causing splatter, and to pressure at low velocities, causing blotches on lines. Whilst the latter is a genuine problem that needs to be addressed in future versions of the tool, the former seems better adjusted as a ‘preference’ by an interface widget, a slider perhaps. The degree of splatter is not a variable that needs constant adjustment, reflecting as it does a personal preference rather than the emotional content of a particular stroke. Adjusting the overall level of splatter with a slider shouldn’t present the same sort of interruption to drawing that adjusting line weight by a similar mechanism causes.



Figure 14 left. A composite image created from live television broadcasts.

Figure 15 right. Fern leaves drawn with the sketching tool after 15 minutes practice by a landscape architect.

7. CONCLUSIONS

The transparent medium opens up the computer interface to ambiguity. Each application sees no change to its environment, and yet the user sees a completely new way to work with the computer. The user can bring together applications with different data models and effectively fuse their capabilities into a new working environment. An application can be exploited, for purposes beyond those defined by its authors, as part of an interface constructed by the user. The sketching tool presented here is just one potential tool for use within this interface.

For a computer to support sketch design it must allow the user to freely combine images and other data, it should promote creation of new images over the editing of existing drawings and it should not interrupt the user whilst she is sketching or otherwise force her to make a decision about the content of the drawing. The transparent medium describes not only the sketching interface, but also the computer's role in the interaction. The computer should literally be a transparent medium.

This research was funded by the Engineering and Physical Sciences Research Council and supervised by Paul Richens. It forms part of a PhD entitled ‘The Potentials of Transparency in Sketch Design Interfaces’ submitted to the University of Cambridge in July 1998.

REFERENCES

- Bier E, Stone M, Pier K, Buxton W, DeRose T, 1993, "Toolglass and Magic Lenses: The See-Through Interface", *Computer Graphics (Proc. SIGGRAPH)* August 1993 73-80
- Bier E, Stone M, Fishkin K, Buxton W, Baudel T, 1994, "A Taxonomy of See-Through Tools", *Proc. CHI 94*, Boston, MA 358-364
- Carter K, 1993, "Computer Aided Design: Back to the Drawing Board", *Proc. Creativity and Cognition 93*, Loughborough
- Colby G, Scholl L, 1991, "Transparency and Blur as Selective Cues for Complex Visual Information", *SPIE Image Handling and Reproduction Systems*, vol 1460 114-118
- Coyne R, 1995, *Designing Information Technology in the Postmodern Age* (MIT Press, Cambridge, MA)
- Gross, M D, 1994 "The Fat Pencil, the Cocktail Napkin, and the Slide Library", *ACADIA '94*, Association of Computer Aided Design in Architecture 103-113
- Gross, M D, 1996 "The Electronic Cocktail Napkin - a computational environment for working with design diagrams", *Design Studies*, 17 (1) 53-69
- Harrison B, Ishii H, Vicente K, Buxton W, 1995a, "Transparent Layer User Interfaces, An Evaluation of a Display Design to Enhance Focused and Divided Attention", *Proc. CHI 95*, Denver, CO 317-324
- Harrison B, Kurtenbach G, Vicente K, 1995b, "An Experimental Evaluation of Transparent User Interface Tools and Information Content" *Proc. UIST 95*, Pittsburgh, PA 81-90
- Harrison B, Vicente K, 1996, "An Experimental Evaluation of Transparent Menu Usage", *Proc. CHI 96*, Vancouver 391-398
- Jansson D, Smith S, 1991, "Design Fixation", *Design Studies*, 12 (1) 3-11
- Kamba T, Elson S, Harpold T, Stamper T, Sukaviriya P, 1996, "Using Small Screen Space more Efficiently", *Proc. CHI 96*, Vancouver 383-390
- Kramer A, 1994, "Translucent Patches: Dissolving Windows", *Proc. UIST '94*, Marina del Rey, CA 121-130
- Kramer A, 1996, "Dynamic Interpretations in Translucent Patches", *Proc. AVI 96*, Gubbio
- Kreuger M, 1991, *Artificial Reality II* (Addison-Wesley)
- Lawson B, 1990, *How Designers Think. The Design Process Demystified* 2nd Ed. (Butterworth Architecture)
- Lieberman H, 1994, "Powers of Ten Thousand: Navigating in Large Information Spaces", *Proc. UIST '94*, Marina del Rey, CA 15-16
- Metelli F, 1974, "The Perception of Transparency", *Scientific American*, April 1974
- Penz F, Bourne M, Wright P, 1992, "Tools for Design: a controlled experiment comparing computer work with traditional hand drawings", *Penz F (Ed.) Computers in Architecture: Tools for design* (Longman, London) 3-14
- Pérez-Gómez A, 1983 *Architecture and the Crisis of Modern Science* (MIT Press)
- Perlin K, Fox D, 1993, "PAD: An Alternative Approach to the Computer Interface", *Computer Graphics (Proc. SIGGRAPH)* Anaheim, August 1993 57-62
- Purcell T, Gero J, 1996, "Design and other types of Fixation", *Design Studies*, 17 (4) 363-383
- Rattenbury K, 1995, "In Glorious Technicolour", *Building Design*, Jan 27, 1995 12-14
- Saund E, Moran, T P, 1994, "A Perceptually-Supported Sketch Editor", *Proc. UIST '94*, Marina del Rey, CA 175-184
- Schofield S, 1994, *Non Photorealistic Rendering: A Critical Examination and Proposed System*, Doctoral Dissertation, Middlesex University
- Schofield S, Richens P, 1995, *The Piranesi System*, (The Martin Centre, Cambridge, England)
- Staples L, 1993, "Representation in Virtual Space", *Proc. INTERCHI 93*, Amsterdam 348-354
- Vesely D, 1993, "Architecture, Simulation and Reality", *Scroope Five*, Cambridge Architecture Journal 5 26-28
- Wellner P, 1993, "Interacting with Paper on the DigitalDesk", *Communications of the ACM* 36 (7) 87-96
- Zhai S, Milgram P, Buxton W, 1996, "The Influence of Muscle Groups on Performance of Multiple Degree-of-Freedom Input", *Proc. CHI 96*, Vancouver 308-315