4O

An Approach to Alternative CAD Interaction via Software Emulation and Speculative Prototyping

Emek Erdolu
Code Lab, School of Architecture
Carnegie Mellon University
Pittsburgh PA USA
eerdolu@andrew.cmu.edu

ABSTRACT

How does the digital interface −CAD drawing space− condition the ways architectural designers work to differ from the ways of work with the paper space? How does shifting from the drafting instruments to the graphic user interfaces and input devices −the keyboard and mouse− reconfigure architectural designers' interaction with drawings? What are the existing ways of CAD interaction and what existing forms of embodied interaction do the keyboard and mouse inscribe? What could be an alternative form that could be situated within this reconfiguration? To discuss these questions, this essay takes *AutoCAD* software as a primary research material and first looks briefly at the history to see a part of the sociotechnical imaginaries and motivations around the software's development. The essay then emulates *AutoCAD 2000* −the software's fifteenth release launched in March 1999− to examine the digital interface of the software and analyzes its' aspects −types of operation, object of interaction, and ways of input and interaction− in relation to the ways they frame architectural designers' work and interaction with the drawings. Based on this analysis of the ways of work and interaction with the work reconfigured by the existing CAD software and hardware, the essay finally introduces a speculative prototype / intervention named 4O, an input device that offers an abstract language of communication and an alternative form of embodied interaction with the CAD software, situated within the software- and hardware-mediated ways of two-dimensional drawing work analyzed. Although the detailed possibilities were not fully exhausted or discussed here, the abstract language of communication and the form of embodied CAD interaction the prototype offers as alternatives open up a potential avenue of exploration of CAD drafting / drawing with hand disabilities.

https://doi.org/10.1145/1234567890

KEYWORDS

Computer-aided design (CAD), computer-aided design (CAD) software, sociotechnical context of computer-aided design (CAD), computer-aided design (CAD) interactions

ACM Reference format:

Emek Erdolu. 2019. 4O: An Approach o Alternative CAD Interaction via Software Emulation and Speculative Prototyping. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1 Introduction

As standard tools within Architecture, Engineering, and Construction (AEC) practice, computer-aided design (CAD) applications work for representation, information integration, and performance assessment in building design and construction processes. Examples include software such as *AutoCAD* geared for two-dimensional drafting and documentation; *3DS Max*, *Rhinoceros3D*, *SketchUp*, *CATIA*, or *SolidWorks* for three-dimensional modeling; *REVIT* or *ArchiCAD* for building information modeling (BIM) all of which has been placed to different degrees and within various professions as the instruments for performing everyday design work. In architectural design, *AutoCAD* has a special place as the software is central in producing the discipline's one of the primary media of information and representation: two-dimensional drawings (Figure 1).

Prior to the use of CAD software in practice, architectural drawings were previously on paper only. Manual drafting instruments such as drawing boards, t-squares, compass sets, and pencils were used to execute the drawing tasks. These instruments had shaped the ways designers interacted with the work. How does the digital interface −CAD drawing space− condition the ways architectural designers work to differ from the ways of work with the paper space? How does shifting from the drafting instruments to the graphic user interfaces and input devices −the keyboard and mouse− reconfigure architectural designers' interaction with drawings? What are the existing ways of CAD interaction and what existing forms of embodied interaction do the keyboard and mouse inscribe? What could be an alternative form that could be situated within this reconfiguration?

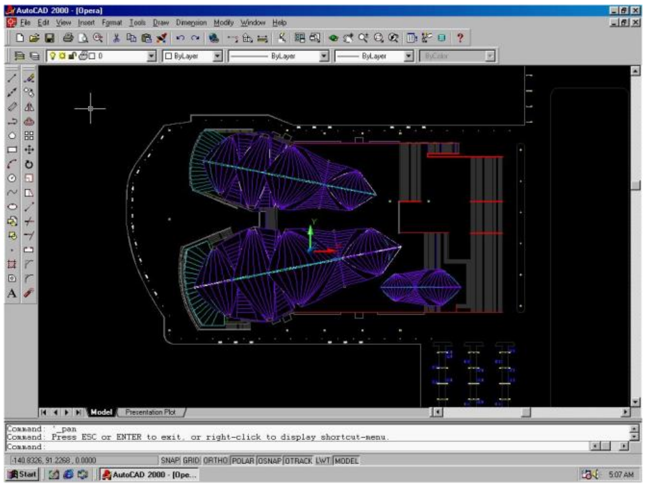


Figure 1: *AutoCAD* 2000 Interface (1999)

To discuss these questions, this essay takes *AutoCAD* software as a primary research material and first looks briefly at the history to see a part of the sociotechnical imaginaries and motivations around the software’s development. The essay then emulates *AutoCAD 2000* –the software’s fifteenth release launched in March 1999– to examine the digital interface of the software and analyzes its’ aspects –types of operation, object of interaction, and ways of input and interaction– in relation to the ways they frame architectural designers’ work and interaction with the drawings. Based on this analysis of the ways of work and interaction with the work reconfigured by the existing CAD software and hardware, the essay finally introduces a speculative prototype / intervention named 4O, an input device that offers an abstract language of communication and an alternative form of embodied interaction with the CAD software, situated within the software- and hardware-mediated ways of two-dimensional drawing work analyzed. Although the detailed possibilities were not fully exhausted or discussed here, the abstract language of communication and the form of embodied CAD interaction the prototype offers as alternatives open up a potential avenue of exploration of CAD drafting / drawing with hand disabilities.

2 Historical Context: An Affordable CAD Software for Design Budget

AutoCAD was developed by *Autodesk*, a small company at the time the software’s first launch in 1982. Initially named Marine Software Partners, the company was founded earlier in the same year by 18 programmers (dropped to 17 soon after) led by John Walker and Dan Drake (Weisberg, 2008). The initial entity was more in the form a programmers’ collaborative, where multiple projects were running as side jobs of its members who were meantime full-time employees for other companies (Weisberg, 2008). Initial set of projects were the ones carried over from *Marinchip Systems Ltd.*, a company that Walker found in 1977 and was operating with Drake since 1980 (Walker, 1989). It included software for a range of purposes such as *Window* for screen editing or *LENS* for lens design (Walker, 1989). There were also new projects running such as *Cardfile* (name changed to *Autodesk* later) for filing (Walker, 1989). So, the focus was not necessarily on design software only.

At the foundation of the software was a graphic program called *Interact* originated by one of the key members of the company, named Michael Riddle, in 1977. As the ancestor of *AutoCAD, Interact* was brought into the company by Riddle, who continued to work on the product[1] (Weisberg, 2008). He and Walker developed the first control program / monitor (CP/M) and IBM PC versions exclusively by converting the software to “C” programming language (Weisberg, 2008). *AutoCAD-80* was featured in *COMDEX*, a popular computer trade show of the time in Las Vegas, in November 1982, and became a big hit as a product but not as big in sales[2] (Walker, 1989). Following for the group was finalizing the *Intel 8086*, *Victor 9000*, and IBM PC version, *AutoCAD-86*, which was released in 1983 and brought sales and some financial success to the company (Weisberg, 2008). The financial success brought by *AutoCAD* changed the entity from a multi-project collaborative into a small company mainly around a single promising product (Weisberg, 2008). Sales of products exploded in 1984, including 1000 copies of *AutoCAD* sold (Weisberg, 2008).

Aside from the foundation of *Autodesk*, the 1980s also witnessed the rise of developments within the commercial desktop systems using input devices like the mouse and keyboard and new screen interaction techniques like the graphical user interfaces and windows (Myers, 1998). *AutoCAD* as a commercial software was sort of a tap into that potential developments in computing that was coupled with decrease in the cost of CAD systems[3] [4] (Weisberg, 2008). The software was fitted to work with these new features of the PC and that brought success that the commercial CAD systems of the past could not get (Figure 2). In software’s first commercial ad on a 1984s *Scientific American*, the capability of running a CAD system on a PC and affordability were highlighted as the two of the main advantages of the software[5] (Figure 3). The vision of

[1] The name of *Interact* was changed to *MicroCAD* soon after and later to *AutoCAD*.
[2] The version was tagged as 80 is because it ran on machines with the *Zilog Z80* or *Intel 8080*.
[3] Some of the earlier CAD systems in the 1970s were also commercially available. However, these systems were using mainframe computers and graphics terminals or minicomputers with a disk drive and one to four terminals (Weisberg, 2008). The providers were hardware manufacturers such as *Applicon*, *Computervision*, *Calma*, and *M&S Computing* without a proper software business model and were closer to the domain of engineering operations (Weisberg, 2008).
[4] CAD systems running on engineering workstations also allowed an increase in performance as graphic manipulation functions and software execution are offloaded from the host computer and left the host to only deal with file management (Weisberg, 2008)
[5] The software was at 2000 dollars sales price back then, and it could be run on a desktop workstation for less than 15000 dollars (1984).

the newer versions of the system running on future computers was emphasized as the way the investment of buying the software would pay for itself.

Figure 2: *AutoCAD* Release 1.0 on *IBM* PC (1982)

Figure 3: *AutoCAD*'s first commercial ad in *Scientific American Magazine* (1984)

The software's features involved a drawing space on which the user can draw / draft, manipulate, and store the geometric representations of designs and a series of command-, text icon-, or tablet-based inputs to do those actions. In 1984s ad, this one-touch command system and choice of input devices – pointing devices, mice, and easy-to-use drawing tablets– suitable based on the drafting needs was reflected as convenient to learn and use (1984). The ad was implicitly suggesting an ease in training the staff and an increase in the speed of drawing / drafting work by drafting in digital with a command-based system (1984). The potential benefit of the new interaction was primarily considered from the perspective of design budgets. Essentially, professional computer-aided design on affordable personal computers via a one-touch command-based software was *Autodesk*'s selling point and resolution to make the design process faster and cost-effective.

3 The Software: Conditioning Architectural Design and the Ways Architects Interact with the Work

For over the years, *AutoCAD*'s capacities with the two-dimensional line-based drawings for representation, drafting, and documentation has emerged as the software's real forte. With that forte, the software has become an industry standard and nearly monopolized the computer-aided drafting in architecture. The software is so widely used and being accustomed that it has become almost synonymous with the acronym CAD, which actually stands for an entire family of software systems. Any faithful user would recognize the application's signature interface that consists of multiple toolbars with words or icons for operation, a text-based command entry bar at the bottom, and the black drawing space in the middle (Figure 4).

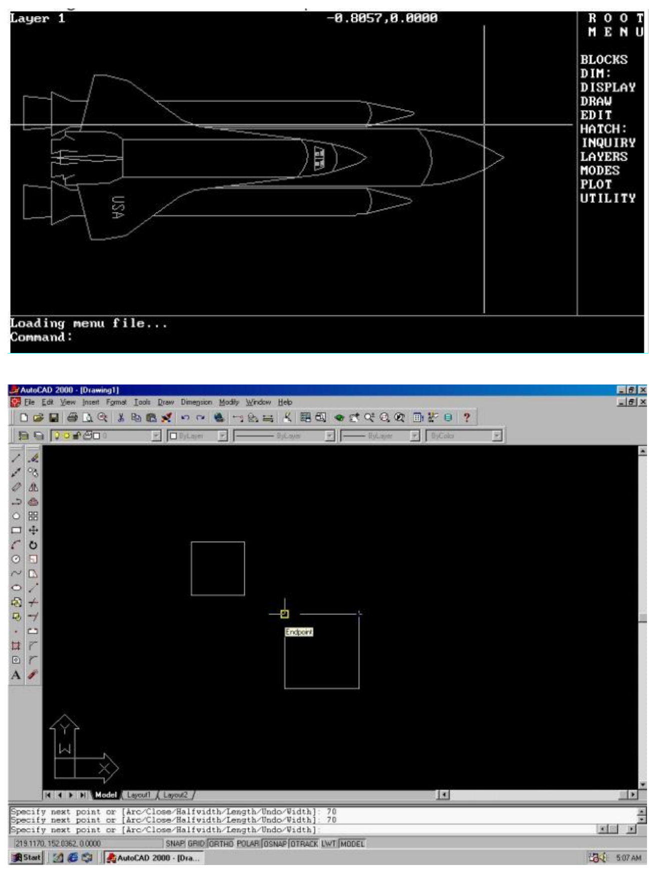


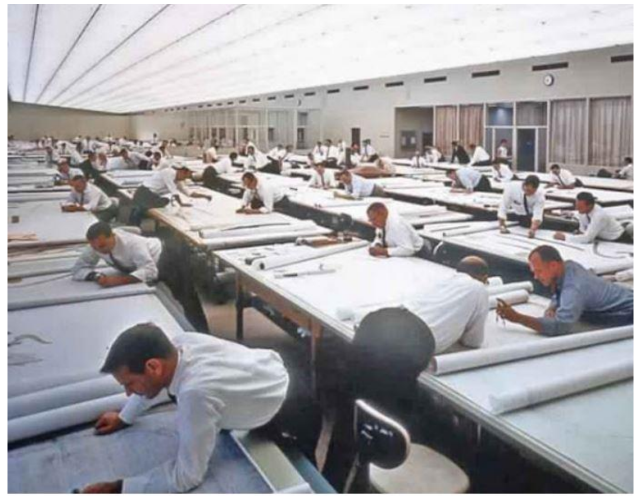
Figure 4: *AutoCAD* Interface –Release 1.2 in 1983 (top) and Release 15 in 1999 (bottom)

AutoCAD is operated through cycles of actions or operations encoded into functions and methods in the back end. For instance, what we say in common language as "copying a geometric object" is an encoded operation that gets called and executed by the software via a user input to the application. Conventionally, user inputs are given via the mouse and keyboard. By clicking on an icon or alternatively by typing in a command, the user can call the functions, and execute the designated steps of it to compose and manipulate geometric objects. Through that (s)he can create abstract or concrete representations of parts, products, buildings, or plans. Besides conventional input via the mouse and keyboard, customizing the encoded operations via *Visual LISP* programming has also become an option since *AutoCAD 2000* (2019). The computation in this case enabled to carry the drafting / drawing process from on paper in physical into a graphic user interface and to execute its essential actions and subtasks by icon- or command-based inputs via the mouse and keyboard instead of the drafting instruments. How did this shift from paper to the graphic user interface, from physical actions to software operations, and from drafting instruments to the input devices condition the ways of work? How did it change the ways architects interact with their work? A critical look at *AutoCAD 2000* and examination of the types of operation, object of interaction, and ways of input and interaction defined by the software may help to explore the two questions on the software's impact.

3.1 Types of Operation

In terms of the types of operation, *AutoCAD 2000* provides three general categories of operations for the user: file management, toggle and screen management, and object manipulation (Figure 5). File management operations involve the commands for managing the files and file properties between the software and directories. Toggle and screen management operations are conducted by the commands for appropriately navigating the screen such as zoom, pan, or orbit. Object manipulation operations contain the essential CAD commands for creating the objects and manipulating their properties.

These three categories of operations in general has given a new form to the way drafting / drawing is performed. A manual process that used to be made of the steps of a technical drawing exercise executed with the drafting instruments on paper became a procedural one that requires activating a command and following its coded steps to execute a drawing action. This new form certainly expanded the possibilities one could do in architectural drafting / drawing such as the ability to manipulate the architectural components quickly and repeatedly, the possibility of duplicating or overlaying the entire parts of a drawings. Aside from the possibilities of quick and endless number of iterations, it also introduced a new level of precision to drafting / drawing practice. With the new form of drafting, scenes of architects and draftsmen working meticulously on drafting tables are replaced by scenes where they sit in front of screens (Figure 5).



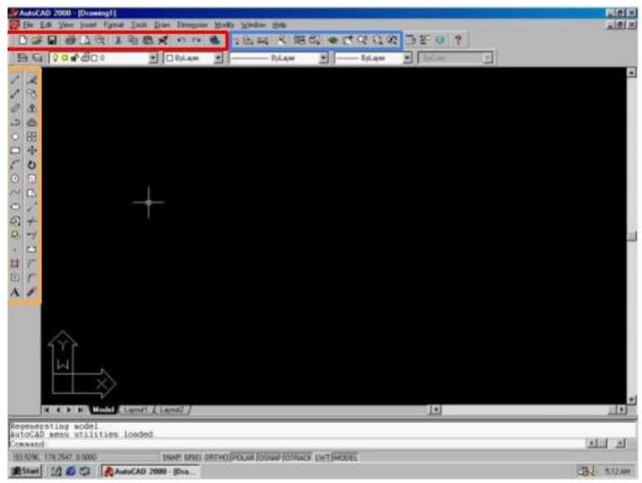


Figure 5: Drafting before CAD in the 1980s (top); file management (in red frame), toggle and screen management (in blue frame), and object manipulation (in

yellow frame) command sets in *AutoCAD 2000* (middle), and a view from BIG's Copenhagen office (bottom)

Each category of operations provides support for drafting / drawing within a particular architectural design process, created new capacities, and conditioned the way files and workflows are organized. For instance, layer / line properties and plot / layout settings (under file management) are central in preparing design or construction documentation (Figure 6). These capacities have not only defined the ways construction drawings are articulated and formatted but also allowed offices to develop custom drawing and layer standards. Another example is using external references that allowed for co-located or remote collaborative drafting work. By external referencing, different design teams and consultants can work on different parts of the same drawing in separate *AutoCAD* files and computers (Figure 7). Any update on an externally referenced part gets automatically updated in the master drawing. New technical and aesthetic standards and forms of collaboration around software were initiated and established by these set of operations.

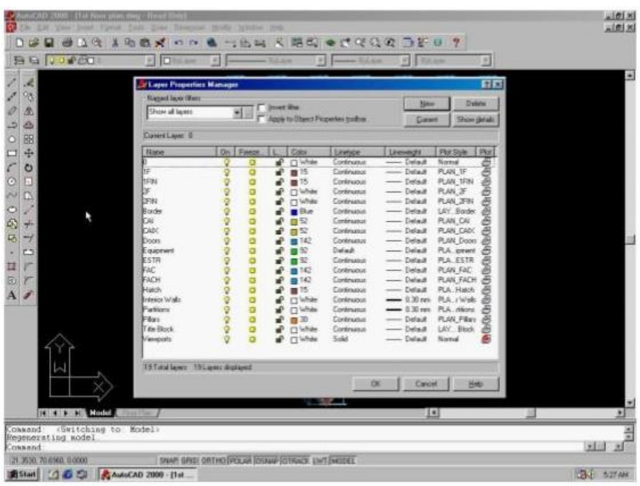


Figure 6: Layer / line properties in *AutoCAD 2000*

Toggle and screen management features are some of the major capacities that transferred the drafting work from the drafting table in physical to the interface. Functions behind these features translated the utilities of the analog drafting instruments like t-rule into digital operations. Working inside an infinite space in *AutoCAD* has allowed designers to work with 1-to-1 scale and with navigating and zooming in / out instead of increasing or decreasing the paper size according to the changing scales (Figure 8). Similarly, the option of (re)orienting the universal coordinates is the function that executes in digital the physical action of rotating the drafting paper or moving around it to change the point of view (Figure 8). A three-dimensional version of navigating and orienting, which is quite hard in actual physical space, is enabled by 3D orbit. 3D orbit combined with zoom options have also allowed to look at the design from high-angle views such as bird's eye that may not readily be possible with a blueprint or a physical model. These operations brought along an entire new set of affordances that were completely out of sight before.

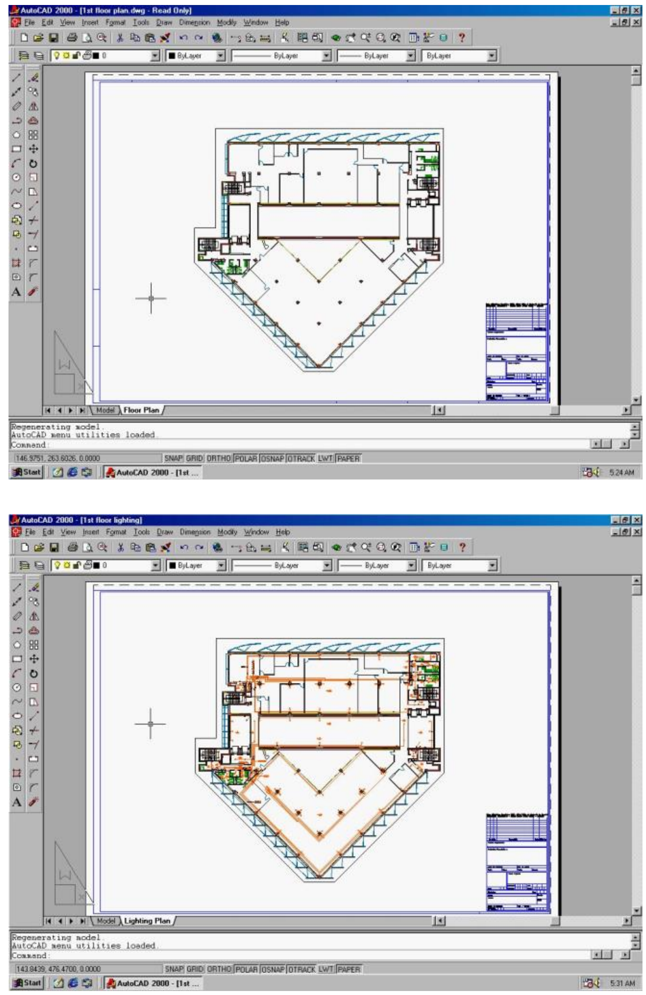


Figure 7: External referencing in *AutoCAD 2000* -Floor Plan (top), lighting systems plan (bottom)

Object manipulation features are the most fundamental. They have allowed designers to create and manipulate geometric objects through a series of repeated commands. As much as working by way of commands helps the designer to speed up or create shortcuts, it also imposes a degree of procedural thinking on designers as they have to think through and take their steps through the order that the encoded operation inscribes. The software defines a custom internal space of operations. In that space, designers deal with the externals of the geometric objects in the order that the back-end code have settled. For instance, mirroring a triangle involves entering the command, selecting the object to mirror, specifying first and second point of mirror axis, and confirm whether to delete the source object or not (Figure 9). Creating a more complex geometric composition involves multiple procedures that the designer has to sort out to find the best possible or the shortest way. These operations were translations of the geometric units and objects, or drawing actions in the software world.

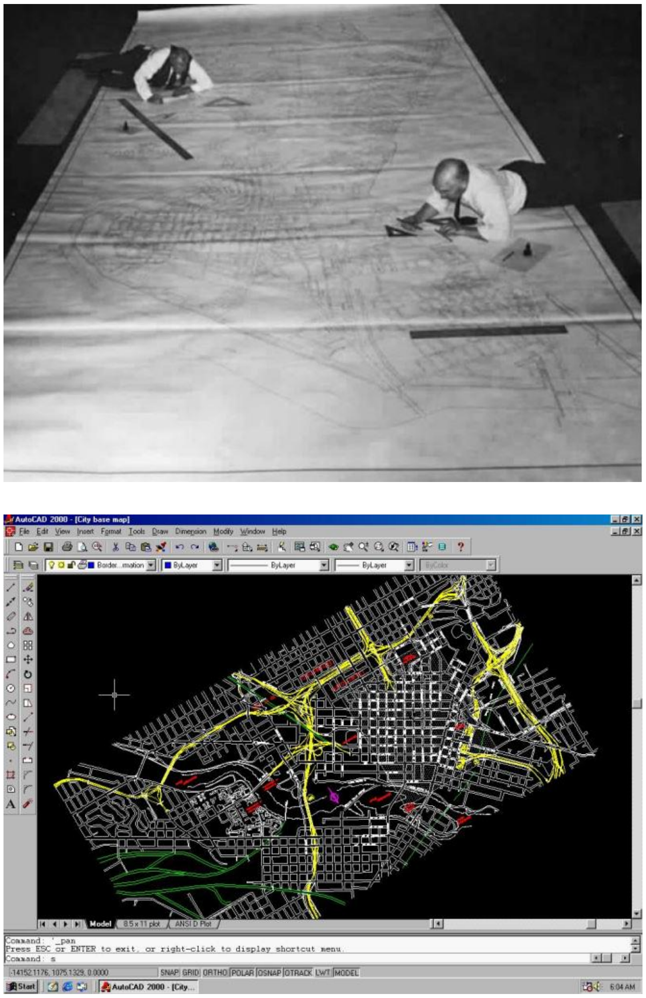
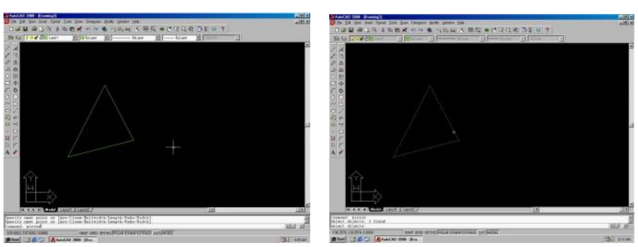


Figure 8: Drafting before CAD in the 1970s (top) and a large-scale drawing navigated in *AutoCAD 2000* (bottom)



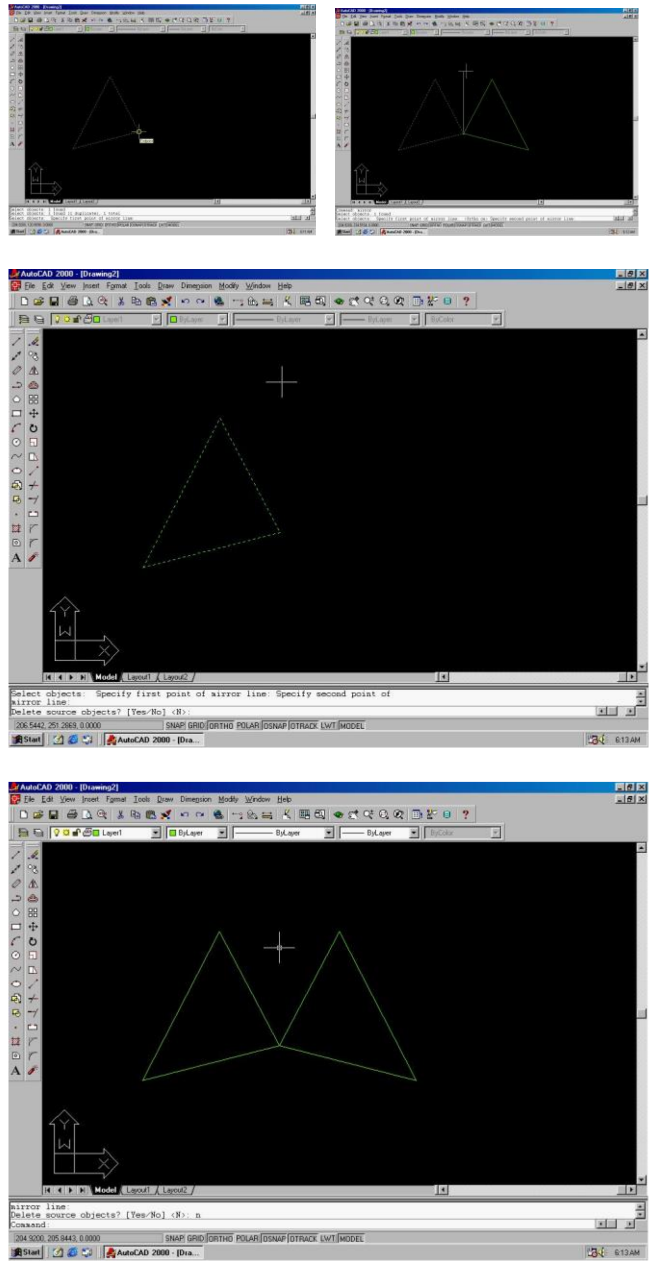


Figure 9: Steps of mirroring an object in *AutoCAD 2000* (in order from left to right and below)

3.2 Object of Interaction

Each type of operation is associated with a different object of interaction. When doing file management operations, the user mainly interacts with the directory and file. In toggle and screen management operations, the object of interaction is the drawing space. With object manipulation operations, the interaction is with the geometric objects and drawing space in digital (Figure 10). Whereas in traditional drafting / drawing, the designer used to interact directly with the drawing in physical (Figure 10. The materiality that designers deal with had changed.

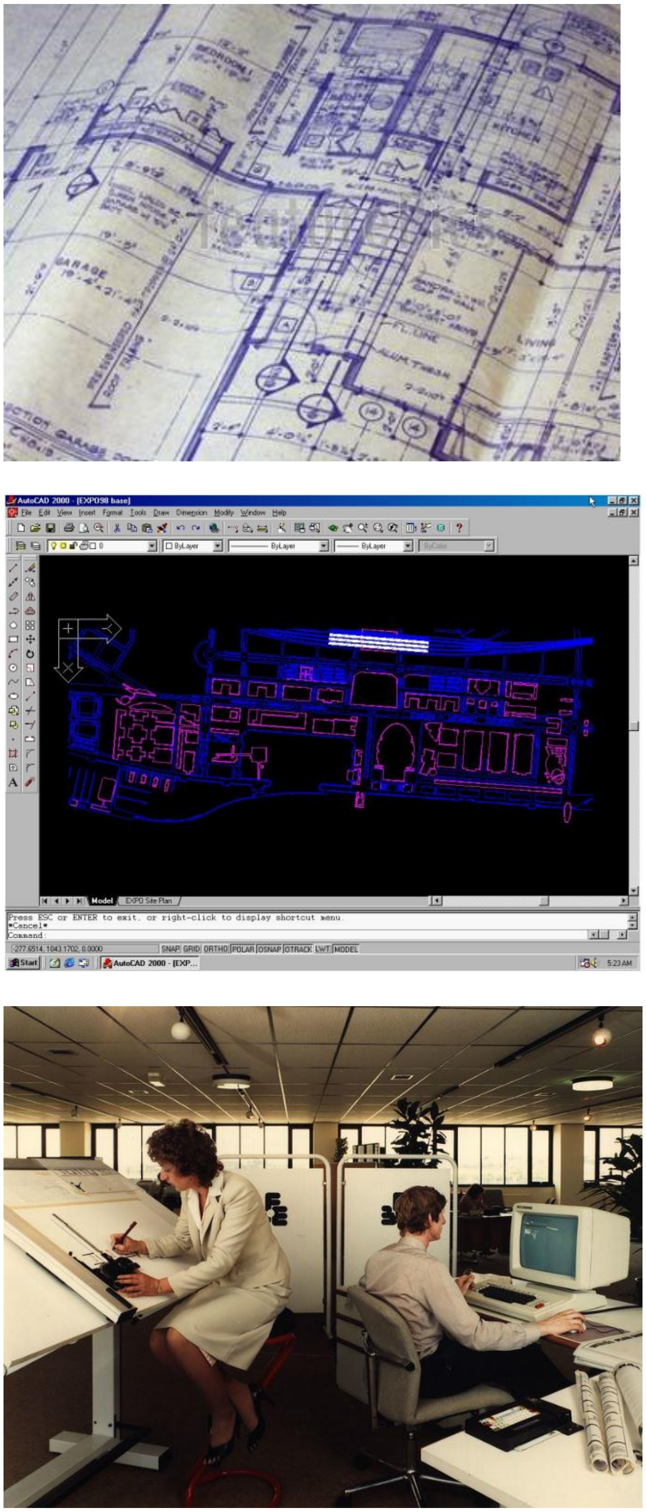


Figure 10: Traditional blueprints as objects of interaction (top); geometric objects and drawing space as objects of interaction in digital (middle), and different bodily interactions by working with the physical drawing versus digital (bottom)

The change in the object of interaction and materiality had subsequently redefined the designer's bodily interaction with the work. Working on a drawing required standing up, leaning over, or moving around the drawing at times, whereas the objects of interaction in digital combined with the screen management features allows and demands sitting down continuously in front of a screen (Figure 10).

3.3 Ways of Input and Interaction

Any command or action under each type of operation could be activated by three alternative ways of input. Activating the command could be via finding and clicking on the text-icon on the Windows toolbar, via clicking on the command icon on the left and top toolbars, or via typing in the command into the command entry bar at the bottom (Figure 11). The traditional drafting / drawing practice included the drafting instruments and direct interaction of the designer with the drawing substance in physical. Whereas the way of interaction had changed when drafting / drawing via the graphic user interface. The drafting work started to involve providing input to execute the designated steps of a command, and the standard input devices –the mouse and keyboard– had become the major instruments as the designer had to constantly interact with the digital drawing space. The designers' motions, which were previously related to drawing, had entirely been redefined by the input devices as the set of motions of interaction with the drawing space.

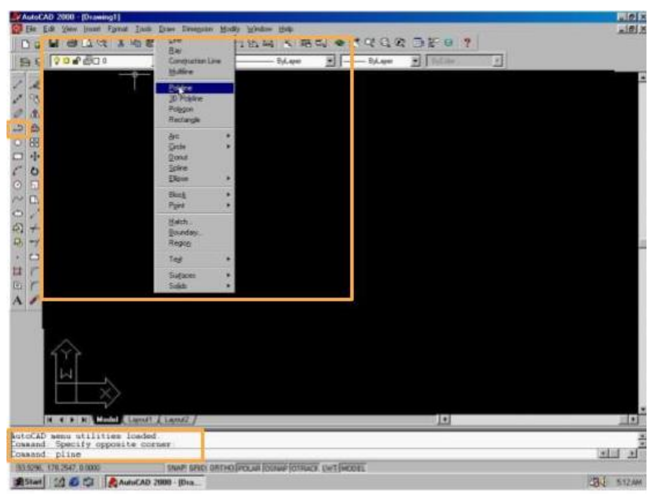
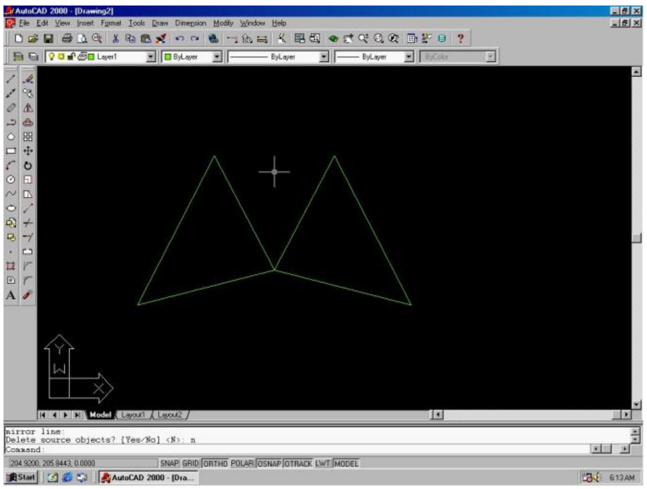


Figure 11: Options of activating commands in *AutoCAD 2000* -text-icon (top), icon (left), and command entry (bottom)

The embodied interactions of the designers while interacting with the digital drawing space are interesting objects of study (Figure 12). No matter in what different weights the designer uses the commands versus icons, the keyboard input is always involved since shortcuts to commands via the keyboard are crucial. If the designer uses type-in commands more, the typical hand-arm configuration would be a fixed left hand on the keyboard, and a right hand in motion between the keyboard and mouse. If (s)he uses icon commands more, then the right hand would remain fixed on the mouse. (Re)thinking about the relations between the internalized operations, type of input, and the way the first two frames tangible interactions with the software opens up an interesting avenue of inquiry on alternative CAD interactions.



\

Figure 12: Arm configuration of a designer while interacting with the drawing space via the keyboard and mouse

4 4O: An Approach to Alternative CAD Interaction

Through the emulation of *AutoCAD 2000*, the essay has discussed the ways CAD software has conditioned the ways of drafting / drawing in architectural design process and reconfigured the ways and instruments of interaction with the materials of the drafting / drawing work. The designer in this new reconfiguration has to work with the encoded operations, with the drawing space and user interface by interacting via the keyboard and mouse, and with new bodily configurations framed by the relation between the user interface, options of commanding, and the two input devices (Figure 13). Maintaining the internal world of operations and interface, yet rethinking the relations between language of commanding, input, and input device could potentially allow to explore and speculate new ways of use, tactility, and tangible interaction in doing drafting / drawing in CAD. A speculative prototype with new forms of interaction could also allow to open up new possibilities for people with hand disabilities to do CAD drafting / drawing.

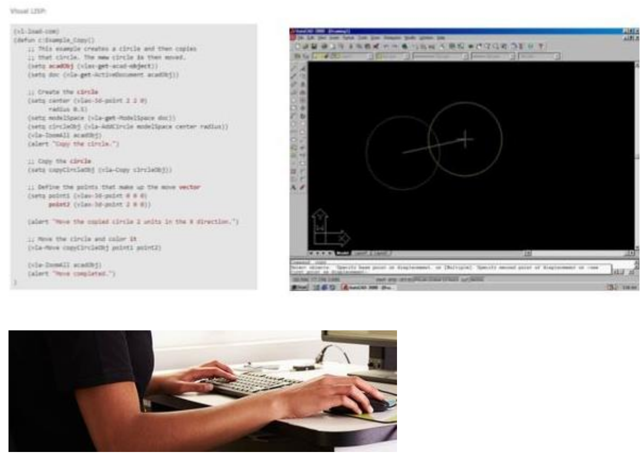


Figure 13: The encoded operations, drawing space and user interface, and input devices reconfiguring the ways and instruments of interaction with the work.

Advancements in graphic user interfaces and input devices within the commercial PC world of the 80s were parallel to *Autodesk*'s efforts in developing *AutoCAD* as a CAD software fitting to those devices (Figure 14). One may say that the features of AutoCAD was designed to work according to the PC's graphic user interface-keyboard-mouse system. And the two input devices has been standard from the 1982's *AutoCAD* Release 1 up to 1999's *AutoCAD 2000* and even now. An interesting way to explore new potential ways of input is to rethink of the language

Figure 14: *Apple Macintosh* (left) and *IBM 3270 PC/GX* (right) ads *Scientific American Magazine* (1984) showing graphic user interface and mouse in personal computers

of commanding, ways of input, and the input itself as representational materialities that communicates the command or action to the software.[6] "What other types of materialities could be used to activate a command?" is the central question here that drove the speculative exploration of alternative CAD interactions.

In command-based CAD software, each operation is represented with a word –line, copy, or move, etc.– or a sort of acronym –*pline* for polyline, *di* for distance, *ro* for rotate– in common language and is activated by a command text entry, icon-click, or a text-icon click. When one activates a command via command text entry, the word is both the representation of the action and the input. The main idea that 4O was developed upon was to translate that representation and input into an abstract visual representation that could also be realized via spatial configurations (Figure 15). And that spatial configuration

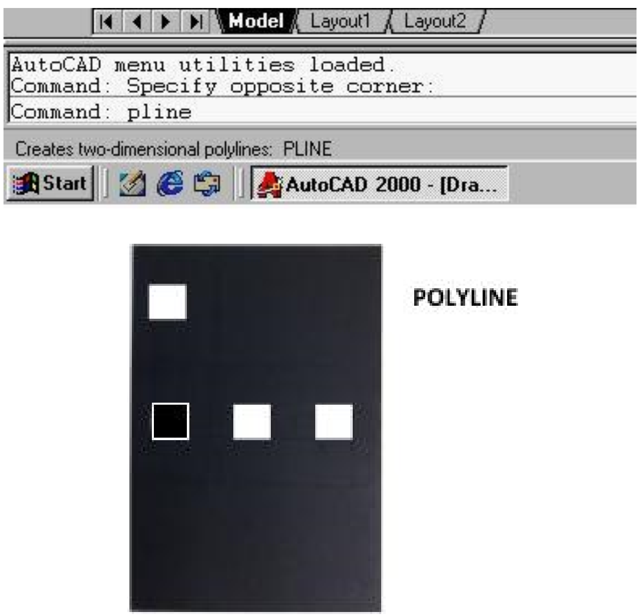


Figure 15: Translating the polyline representation and input into an abstract visual representation that could later be physicalized

could be the translation of representation of command / action and input to the physical. In an abstract fashion, one could imagine that would be various configurations by 1s and 0s on a grid or by blacks and whites in a group of objects. For 4O, that translation involved three holes (or geometric objects) with a fixed position in space that could either be filled (black) or open (white). Each variation of the configuration –filled (black)-open (white)-open(white) or filled (black)-filled (black)-open(white), etc.– would represent a command and the realization of that abstract configuration in physical would provide the input to activate the command.

That conceptual idea was developed into a functioning hardware prototype with a microcontroller (Arduino Uno) and photocells (light sensors). Four photocells were laid out and sandwiched between a 3D printed case and a foam board surface with four openings (4Os) each of which exposes a light sensor (Figure 16). A threshold for the analog values of sensors were set in programming them as switches, and the switch values were used to represent the abstract configurations of blocked (black)

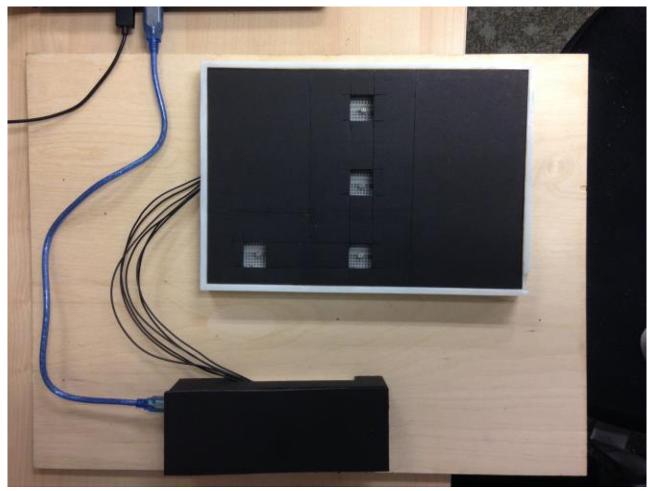


Figure 16: From abstract representation to hardware

or exposed (white). The prototype was interfaced with Rhinoceros 3D via Firefly / Grasshopper, and configurations were programmed in GH Python to activate drafting / drawing commands in Rhino. Six commonly used two-dimensional drafting commands –polyline, curve, trim, copy, mirror, and offset– were represented by singular and twin combinations of the three of the sensors (e.g. first sensor blocked – polyline, or first and second sensor blocked – copy, and etc.) (Figure 17). The fourth sensor was used as a switch from singular to twin combination mode so the twin mode would prevent a single sensor from activating a command and stopping the analog read.

For blocking or opening the sensors in physical, four objects in rectangular prism form were produced. By sliding or moving these objects on the surface, the designer could block and expose the sensors and create the intended configuration to activate the command in Rhino space (Figure 18). This move for calling the command takes what used to be the designer's keyboard hand (Figure 19). And the rest of the operation is

[6] Materialities of information representation or representational materialities refers to an approach to study social phenomena by examining and interpreting the relations between material forms of digital data representation and social action (Dourish, 2017). An example to demonstrate different material forms of data could be a command of action in a software that is possibly communicated to the machine by various formats of input such as a mouse click, a command type, or a piece of code. Another example could be the properties of information representation and organization across different software.

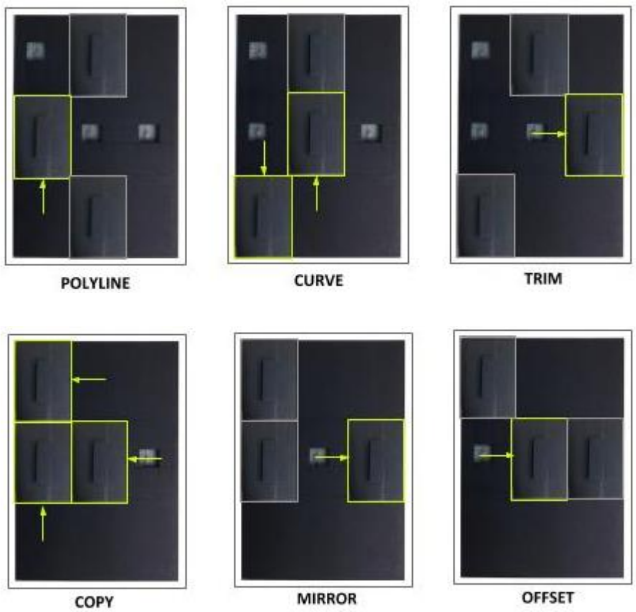


Figure 17: Representations of six commonly used commands by the 4O system

executed with the mouse hand by using left click to interact with the drawing space and right click to enter the finished steps of a command (Figure 19). Rectangular prism on a surface in comparison to a button brings about a new set of affordances, introduces different forms of tangible interaction, and flexes the precision required with the keyboard buttons; all of which open up new possibilities of use without the constraints of holding, grasping, or finger-pushing or with customized prosthetics. These possibilities could be further explored for expanding the CAD interaction for people with hand disabilities.

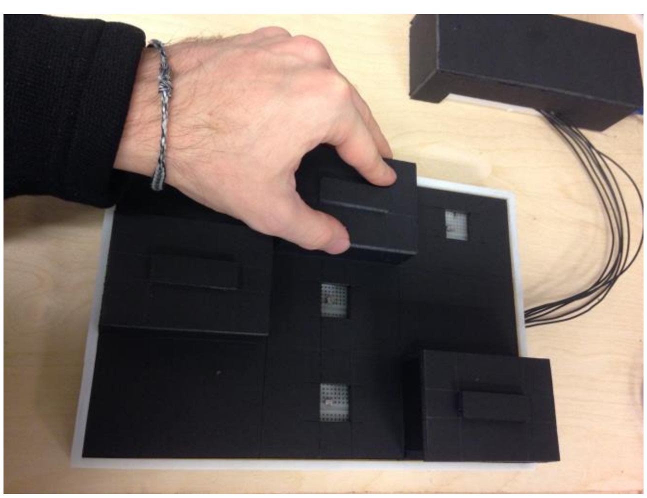
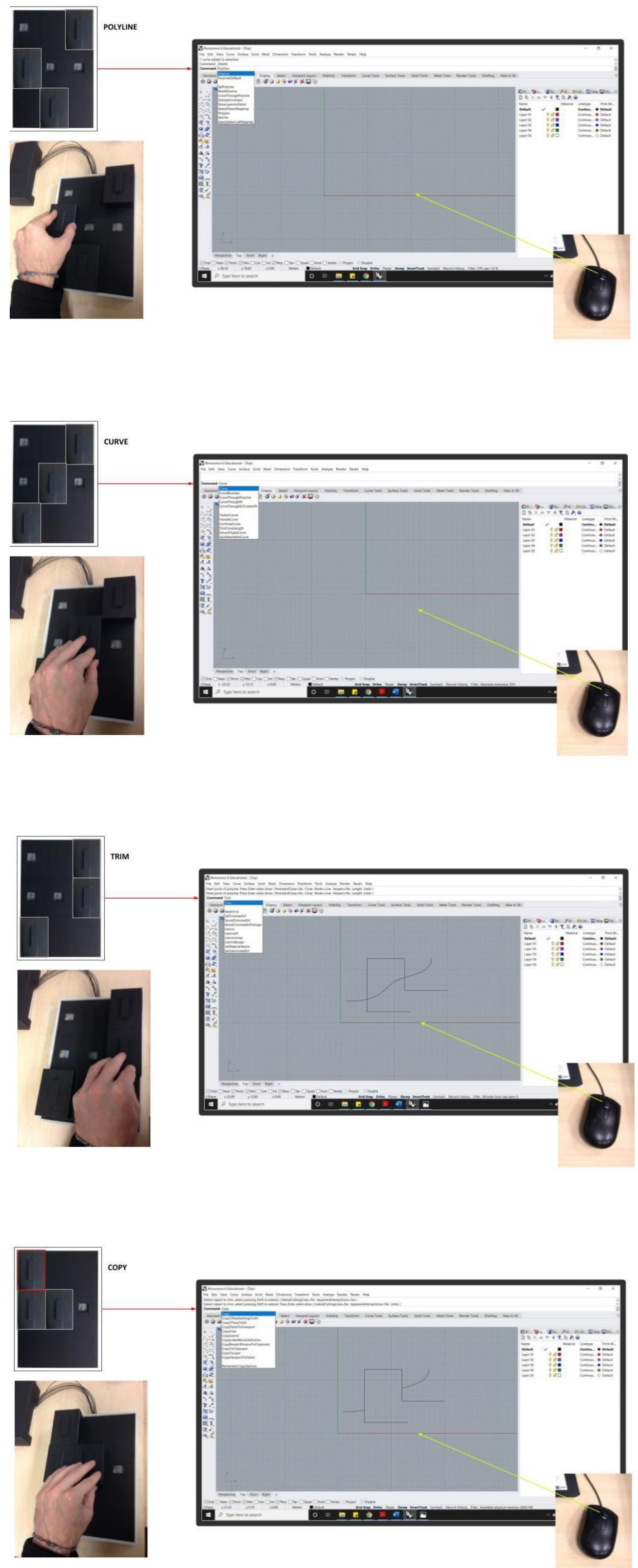


Figure 18: Blocking or exposing the sensors by sliding or moving these objects to activate the command



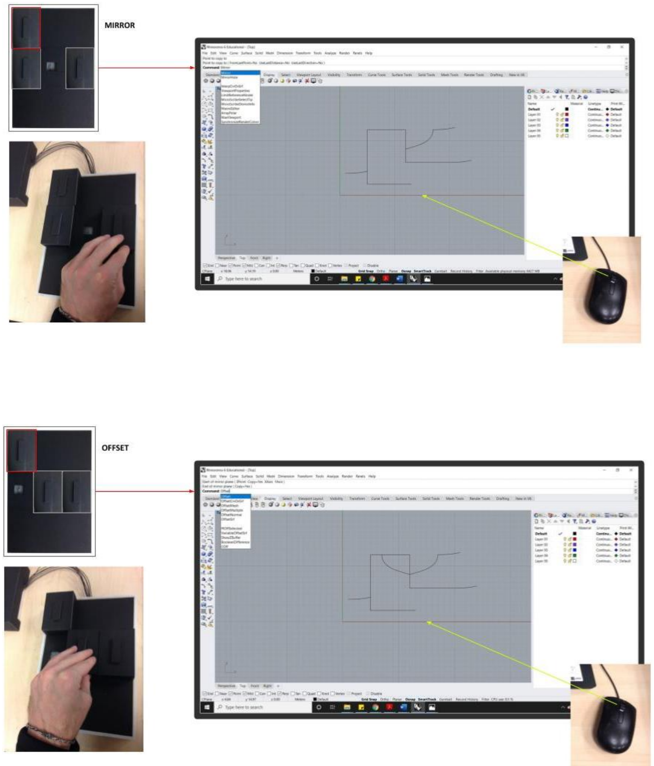


Figure 19: Left hand moving the objects on 4O for calling the command -polyline, curve, trim, copy, mirror, offset (from top to bottom)- and right hand with the mouse for executing the rest of the operation

5 Conclusion

Through a brief look into history, software emulation, and speculative prototyping / intervention, this essay has discussed a few key questions on the ways CAD software and hardware as a drafting / drawing instrument has been reconfiguring the work and the designer’s interaction with the work, and has explored different materialities in designing an alternative CAD input / interaction situated within this reconfiguration. The extent and depth of analysis on what the essay outlines as –types of operation, object of interaction, and ways of input and interaction– is by no means extensive yet provided a few good ingredients for exploring and making of an alternative input / interaction device as a material of discussion. The abstract language of communication and the form of embodied CAD interaction the prototype offers as alternatives point to potential next steps of exploration of the system as an assistive technology.

REFERENCES

[1] Afralisp.com (2019). Visual LISP: What is Visual LISP?. [online] Available at: https://www.afralisp.net/visual-lisp/ [Accessed 28 October 2019].
[2] Autodesk (1984). AutoCAD Ad. In *Scientific American* Vol.251 No.3, Scientific American, Inc., pp.225–226.
[3] Dourish, P. (2017). *The Stuff of Bits: An Essay on the Materialities of Information.* MIT Press.
[4] Myers, B. A. (1998). A Brief History of Human Computer Interaction Technology. *Interactions*, 5(2), pp.44–54.
[5] Walker, J. (ed.) (1989). *The Autodesk File: Bits of History, Words of Experience.* New Riders Publishing.
[6] Weisberg, D.E. (2008). Autodesk and AutoCAD. *The Engineering Design Revolution: The People, Companies and Computer Systems that Changed Forever the Practice of Engineering*, pp.8-1–8-51.

Figure 1: (eaasi-prod.library.cmu.edu)
Figure 2: (scan2cad.com/tips/autocad-brief-history/)
Figure 3: (Scientific American)
Figure 4: (autodesk.blogs.com/between_the_lines) (eaasi-prod.library.cmu.edu)
Figure 5: (vintag.es) (designboom.com) (eaasi-prod.library.cmu.edu)
Figure 6: (eaasi-prod.library.cmu.edu)
Figure 7: (eaasi-prod.library.cmu.edu)
Figure 8: (vintag.es.) (eaasi-prod.library.cmu.edu)
Figure 9: (eaasi-prod.library.cmu.edu)
Figure 10: (featurepics.com) (eaasi-prod.library.cmu.edu)
Figure 11: (eaasi-prod.library.cmu.edu)
Figure 12: (thumbtack.com) (eaasi-prod.library.cmu.edu)