Negroponte, N. (1972). HUNCH - An Experiment in Sketch Recognition. In Environmental Design: Research and Practice, Proceedings of the EDRA 3/ar 8 Conference, University of California at Los Angeles.

22: MAN-MACHINE SYSTEMS

HUNCH
AN EXPERIMENT IN SKETCH RECOGNITION

The Architecture Machine Group
Department of Architecture
Massachusetts Institute of Technology

Nicholas Negroponte
Leon B. Groisser
James Taggart

This paper highlights the operations and problems of a set of computer programs called HUNCH. The specific goal of HUNCH is to be able to recognize and make inferences about a user's sloppy, incomplete and equivocal drawings without that user having to be more explicit or categorical than he would be with an onlooking human colleague. Steps toward this goal are being made within a particular context -- architecture -- that furnishes a "knowledge base" or "assumption base" from which programs can procure (rather than develop) those heuristics necessary to handle two dimensional and three dimensional ambiguities.

While HUNCH is indeed an exercize in "computer graphics", we believe that a great many of its operating principles are characteristic of all kinds of future systems that will be expected to cope with vague information and to deal with nebulous ideas. For example, 1) we not only expect the program to make errors, but also to benefit from errors (ie: to improve its recognition ability and context handling ability); 2) we expect the program to support an evergrowing acquaintence with the user (ie: a knowledge of his ideosyncracies, habits, whims); 3) we anticipate that the program will have many channels of access to the real world (to the extent of recognizing gestures and smiles); 4) we insist that the program puruse a multiplicity of candidate "solutions", sorted and selected by "criticizer" programs.

Since 1967, the authors[1] have argued that computer-aided design is an issue of artificial intelligence, that machines have to understand the meaning of decisions in order to be partners in making them. We have advocated the exploration of artificial intelligence and have condemned computerized solution generation which does not consider it. One severe consequence of this position is that there remains very little to work on both because of the state of the art of computer sciences in general, and because intelligence per se cannot be tackled in parts.

Sketch recognition has provided us an interim domain. While it is certainly difficult to shift between wanting to deal with "partially formed ideas" and wanting a program to find corners (which is what most this paper is about), it is a topic that has inherent uncertainties and ambiguities at all levels. It is representative of a class of problems that can be tackled with modesty, but in the ultimate case, require an intelligence.

Acknowledgements
Precursory work commenced in 1966 in a project called URBAN5 jointly sponsored by IBM and MIT. Actual work, however, started only in early 1970 with the Bachelor of Architecture thesis of Mr. James Taggart. This work was conducted within the Architecture Machine Group at MIT and was supported by the Ford Foundation. The more concentrated effort began in the summer of 1970 with a grant from the Graham Foundation for Advanced Study in the Fine Arts and major support from MIT's Project MAC (Advanced Research Projects Administration contract N0014-70-A-001).

At present HUNCH is being developed as an integrated package to act as a front end for any system for "Computer Aids to Participatory Architecture",[2] sponsored by the National Science Foundation.

Along with these many generous sponsors, the authors would like to thank Dr. Gordon Pask who, in spring 1971, provided radical inputs, which guided the project into new directions.

Introduction
We view the problem of sketch recognition as the step by step resolution of the mismatch between the user's intentions (of which he himself may not be aware) and his graphical articulations. In a design context, the convergence to a match between the meaning and the graphical statement of that meaning is complicated

by continually changing intentions that result from the user's viewing his own graphic statements. In some sense, the sketch acts as a form of memory for the user while he loops, so to speak, into the real (physical) world to gain a better understanding of his problem. Consequently, the nature of his drawing (ie:the wobbliness of the lines, the collections of overtracings, the darkness of the lines) has important meanings for the most part overlooked in computer graphics. We have already proposed that: "A straight line 'sketch' on a cathode ray tube could trigger an aura of completeness injurious to the designer as well as antagonistic to the design."[3]

HUNCH tries to take into account some usually overlooked (or not taken seriously) graphic subtlties such as: speed of drawing, pressure upon the pen, and crookedness of lines. At present, this is for the purpose of making those transformations necessary to pass from a drawing that is meaningful to the user (figure 1) to an interpretation (figure 2) which is both managable by the machine and appropriate for first approximations, guesses and extrapolations. Later, we expect to employ graphical innuendoes for the purpose of having the machine recognize attributes which have to do with the user's attitude toward his own project, his confidence in a solution and the like.

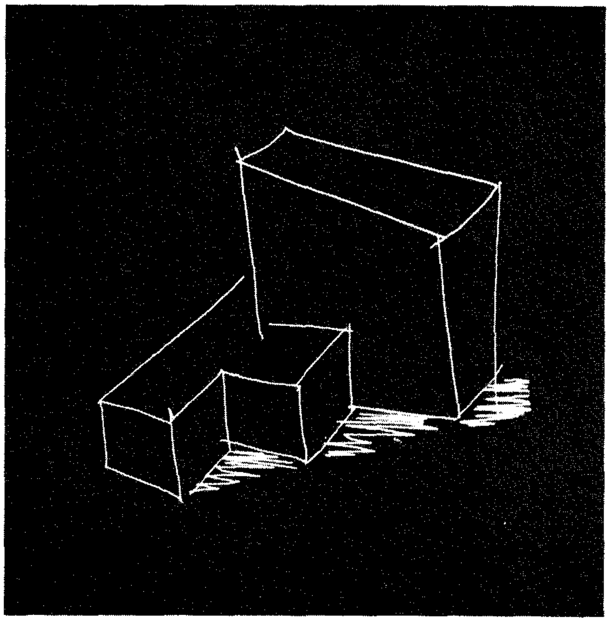


Figure 1.

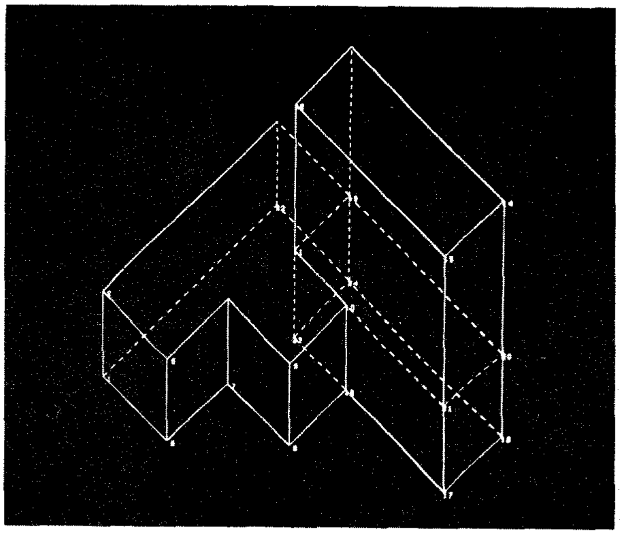


Figure 2.

We believe that HUNCH will have general applicability as a front end to computer systems that require the graphical input of ideas which are not well formed. It should have particular importance for architectural applications inasmuch as present day computer applications are always hampered by the means of imput. The architect interested in computer-aided design techniques must either 1) do a design away from the machine, and at some level of completion initiate the usually clumsy procedures necessary to make it readable by the machine; or 2) must stymie his own design behavior and subject himself to drawing techniques which are inappropriate for "creative" thought. Neither is suitable.

Within the next two years, HUNCH is expected to handle true sketches (figure 3), drawings marked by inaccuracies, missing information and even coffee stains. We propose to do this by: 1) providing the user with a broad range of amenable hardware interfaces with the machine; 2) developing programs that get to know the particular user better and better; 3) creating an inference-making system that capitalizes upon a history of encounters with many users as well as with the particular user.

The following pages should be viewed as an interim report.

Configuration

We start with a description of the configuration of our hardware because we believe that each device has unique operational characteristics built into it.

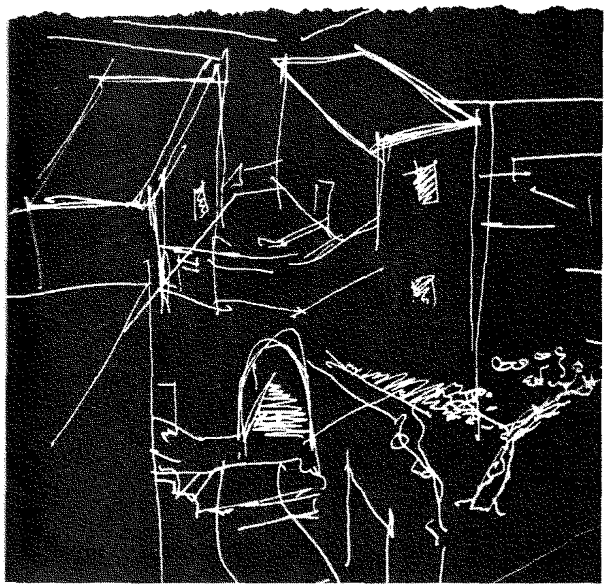
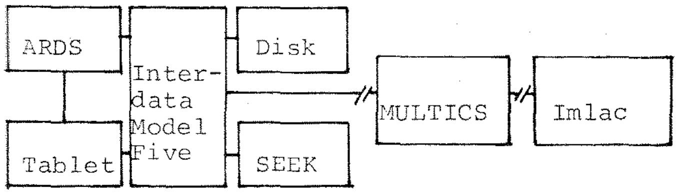


Figure 3

We depend very heavily upon these inherent features for the purpose of recognition as well as of intensive, congenial interaction. The hardware described is a subset of what we call "The Architecture Machine" (figure 4).

The diagram illustrates the subset of The Architecture Machine used for HUNCH as well as the interconnections among the seven operating elements: a Sylvania Tablet, an Advanced Remote Display Station (ARDS), an Interdata model 5 minicomputer, a Diskstor



disk, the MULTICS time-sharing system, an IMLAC display unit and a three dimensional positioning device called SEEK. The reader should note that this configuration is highly redundant and, while it employs a plurality of devices, it is marked by a very low cost, approximately $40,000 purchase price.

Sylvania Tablet

The Sylvania Tablet, with its stylus, is the primary input medium of the system (figure 5). The tablet operates in a manner that is crucial to HUNCH. It issues to the minicomputer a constant number of X and Y (and even Z) coordinates per second (in our case 200 hundred per second). This constant rate acts as a form of a clock.

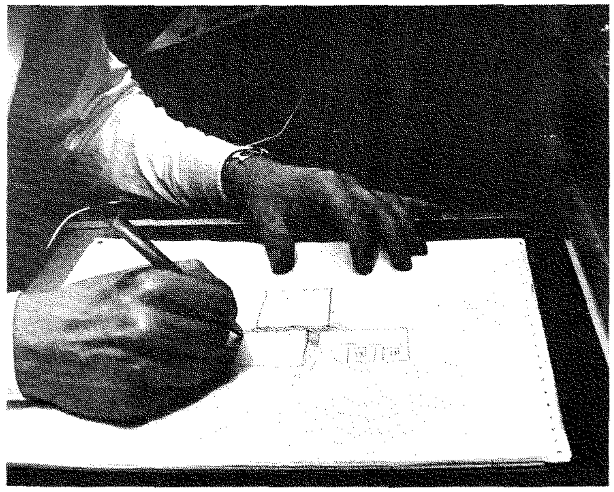


Figure 5

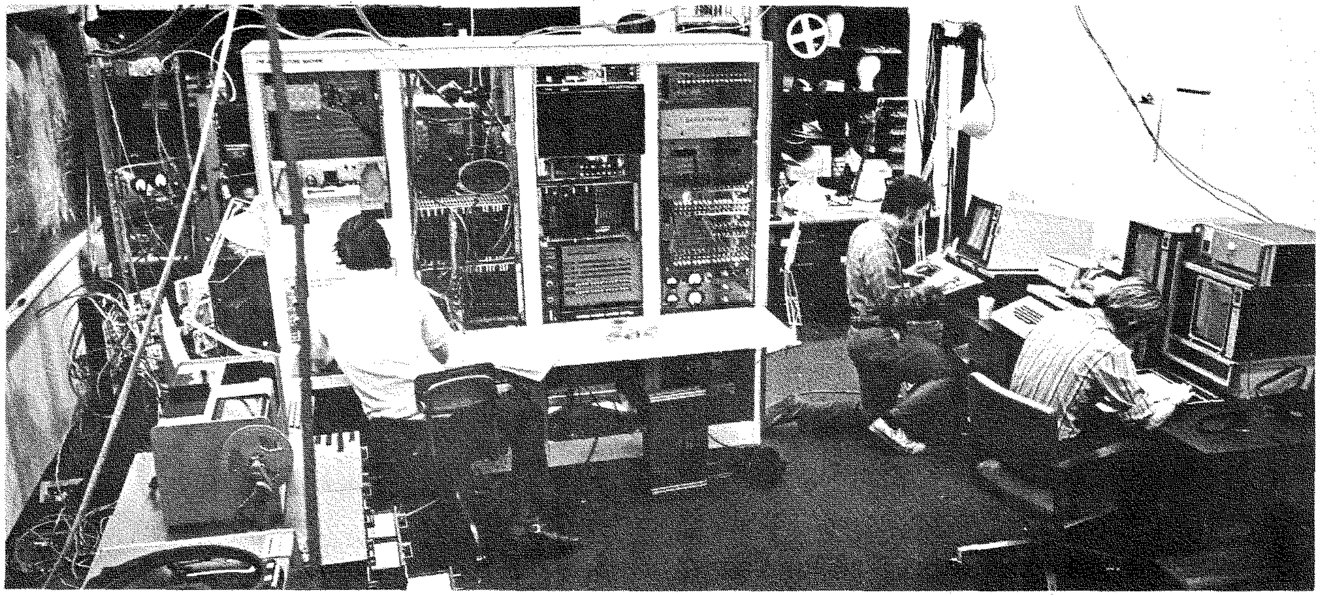


Figure 4

It means that lines drawn slowly will have digitized points closely spaced, whereas lines drawn rapidly will have them more dispersed. As a result, we have a built-in record of the speed at which each part of a drawing is created. We subsequently employ this parameter as a major criterion for determining the user's graphical intentions: "did he mean this to be ... a straight line, a square, a corner, etc?". In effect, we are correlating the user's speed of drawing with his purposefulness.

We also associate intentions with the user's pressure upon the stylus. This feature has been added to the Sylvania Tablet by Wade Shaw as part of his Electrical Engineering Thesis, Textural Input and Definition for Graphic Display. A pressure sensing gage placed within the shaft of the pen measures how hard the user is pressing down. When connected with the focus control of the display, this feature simulates pushing harder on a pencil to get a blacker line. We believe that this is particularly important to sketching because it is quite common to overtrace lines, and thus without erasing, to make the most up-to-date lines emphatically dark.

It is important to note that both speed and pressure provide inputs which would not really be available to an onlooking human, especially if he were looking at the sketch after the fact. Some argue that it is more appropriate or meaningful to artificial intelligence to use only those cues available to humans. We don't.

The tablet has two further important features: First, it has a transparent surface which permits it to be used vertically in front of the display with the minor inconvenience of a half inch paralax (figure 6).

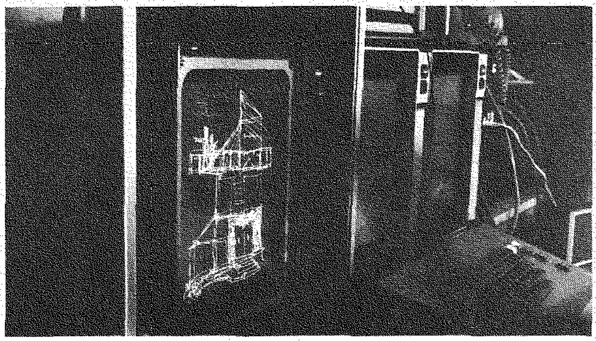


Figure 6

This provides an ability for drawing without the computations necessary for pen tracking (which is necessary for light pens). Second, it has a limited three dimensional capability because it can record three levels of Z (where the range of each level is variable with a screw driver). At this time we make only limited use of this last feature (for example, we determine the probable beginnings and ends of pictures and opportune moments to write information on the disk).

Advanced Remote Display Station

ARDS was the first storage tube system developed (figure 7). The advantage of a storage tube is that it maintains the image on the face of the scope without "refreshing" it and thus without the associated computations necessary to continually redraw the image. (Hence its convenience to time-sharing.) Its drawback is its resulting inability to locally erase, that is to remove a part of the picture without erasing the whole and redrawing it.

Figure 7

This does not pose a problem in our context; consider whether you can locally erase with a felt tip pen, or how often you erase pencil lines. Furthermore, a storage tube allows an unlimited number of vectors to be displayed (in contrast to the more expensive "dynamic displays" that allow only a few seconds of sketching before flicker sets in because so many points must be continually drawn).

Storage tubes have the ability to "write thru". This mode of operation can be initiated under computer control. It affects a low beam voltage which produces

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.