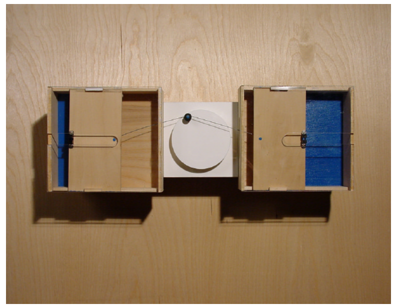
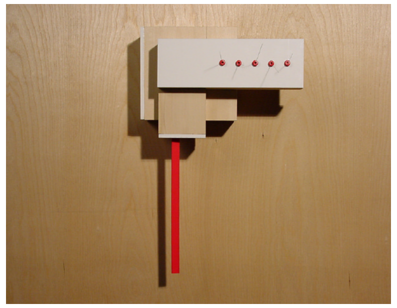
Since June 1999, I have been working and studying in the Aesthetics and Computation Group at the MIT Media Laboratory. During this time, my work shifted from experiments in interface and information design to my current work with computational kinetic sculpture. The common thread in this work is the study of dynamic reactive systems that receive and process input as a means of generating and altering visual compositions. The systems I have been creating are computer programs written in C and Java that run on both high-end computers and microprocessors. I began creating purely digital screen-based compositions using a mouse and keyboard as input and am currently creating physical sculptures which use physical sensors such as cameras, microphones, and touch sensors to receive data from their environment.

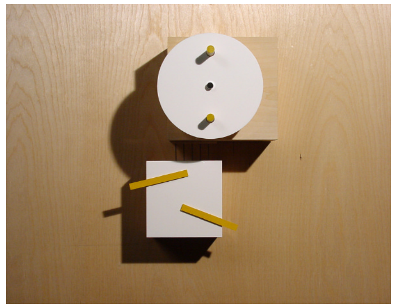
These projects are interactive works of software and hardware. Their meaning is elucidated through interaction and their form is understood through motion. The representations in this book therefore only suggest the actual projects and allude to the experience of interacting with them.

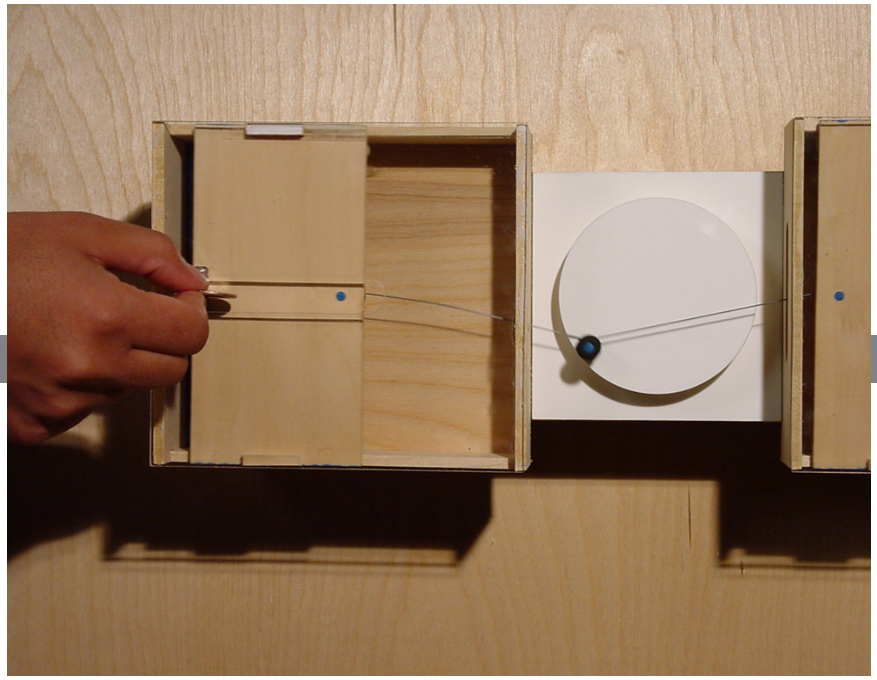
reactive boxes

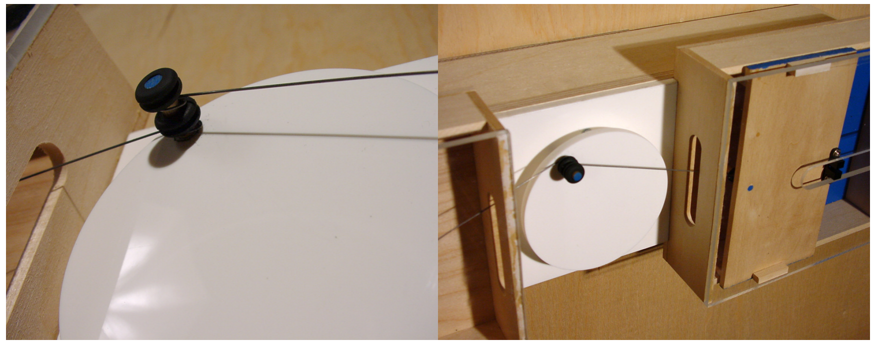
http://acg.media.mit.edu/people/creas/box







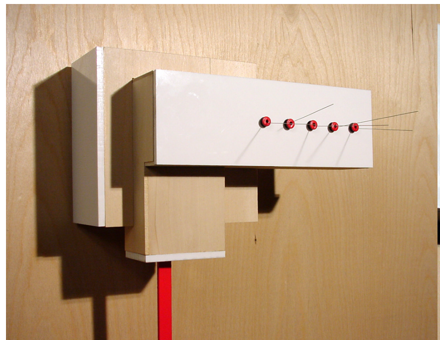
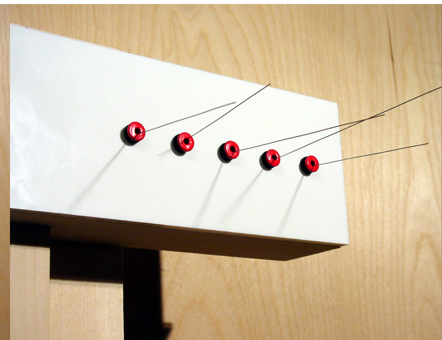
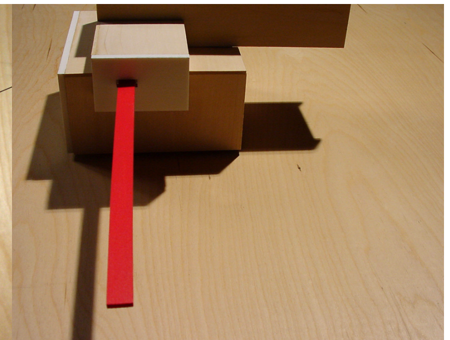
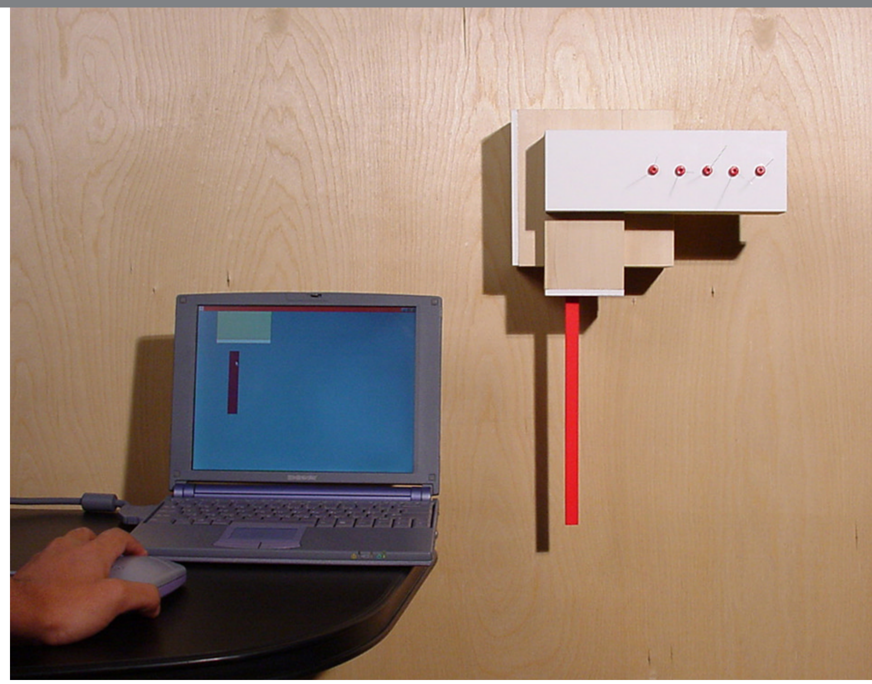
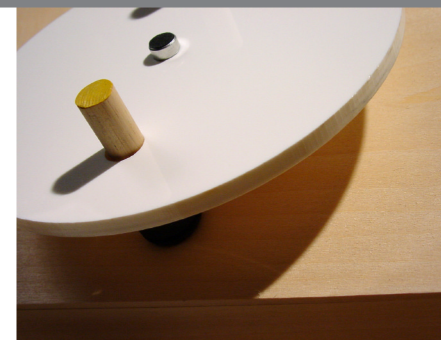


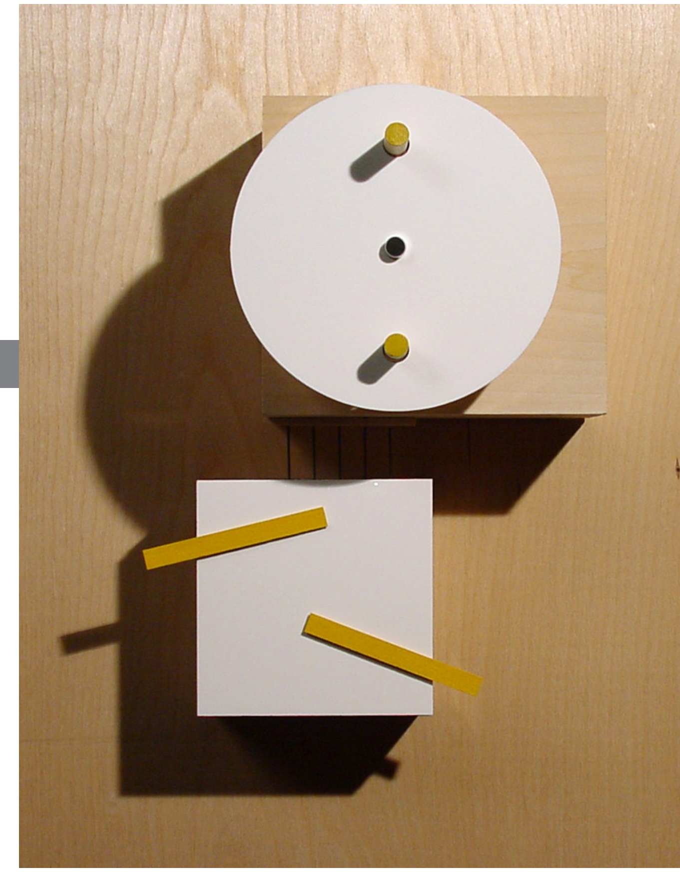


As a precursor to developing larger computational sculptures, I devised a series of three studies to explore various ways of interacting with sculptural systems and to explore different types of sensors and motors. The first study is physically reactive and its motion is triggered by touching one of two sensors mounted on its surface. The second study has a remote interaction via a screen-based interface. Sending a stimulus triggers a solenoid and the temporal difference between stimuli is calculated to modulate the speed of a DC motor which turns a series of wires. The third study has an ambient reaction and uses the sound and light levels in the room to generate motion.

Box One: Physical Interaction

Interacting with the physical box involves triggering one of two sensors mounted to the surface of the object. Triggering the right sensor causes the motor to turn clockwise while triggering the left sensor causes the motor to turn counterclockwise. Triggering both sensors at the same time causes the motion to stop. The electronics used for this box are all analog.

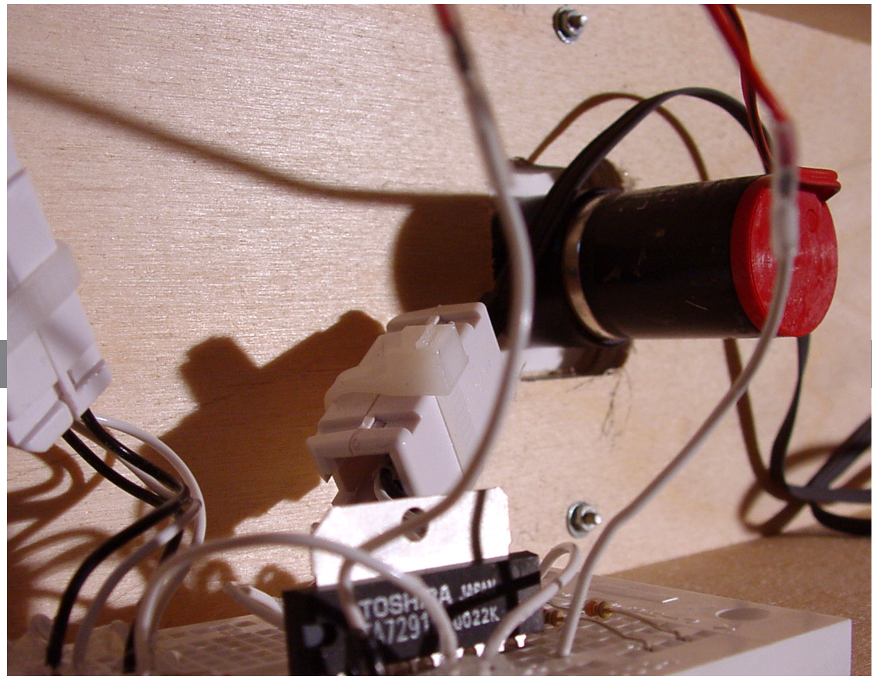


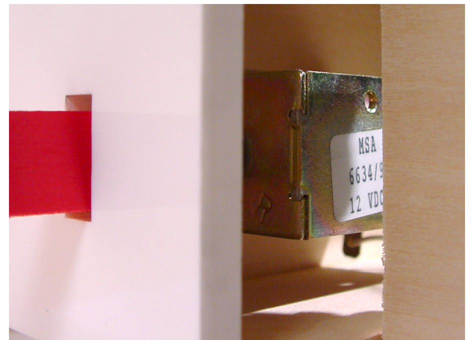
Box Two: Remote Interaction

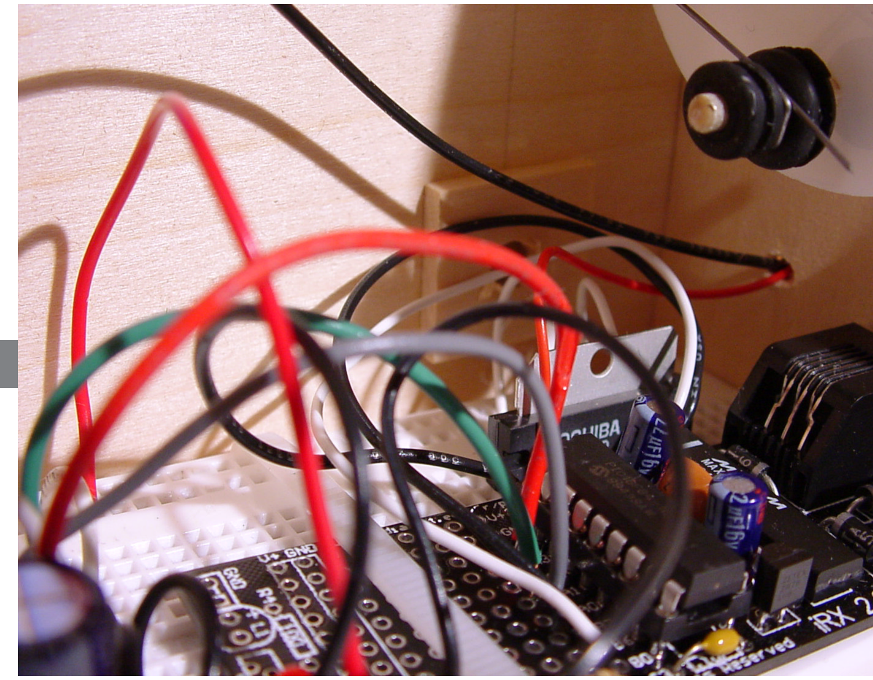
The movement of the remote box is controlled by a microcontroller and the interaction is through a networked java application. A person can send a stimulus to the box by manipulating an abstract digital interface on a computer screen which sends a signal over a computer network. This stimulus triggers a motor inside the box and the time difference between signals determines the speed and rhythm of another motor which turns the wires on the face of the box.

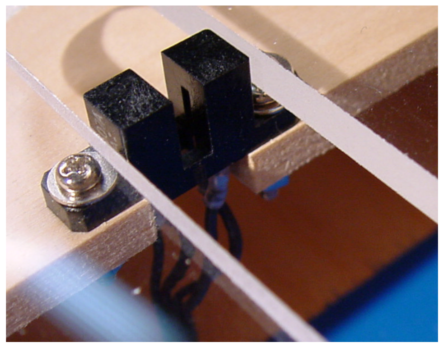
Box Three: Ambient Interaction

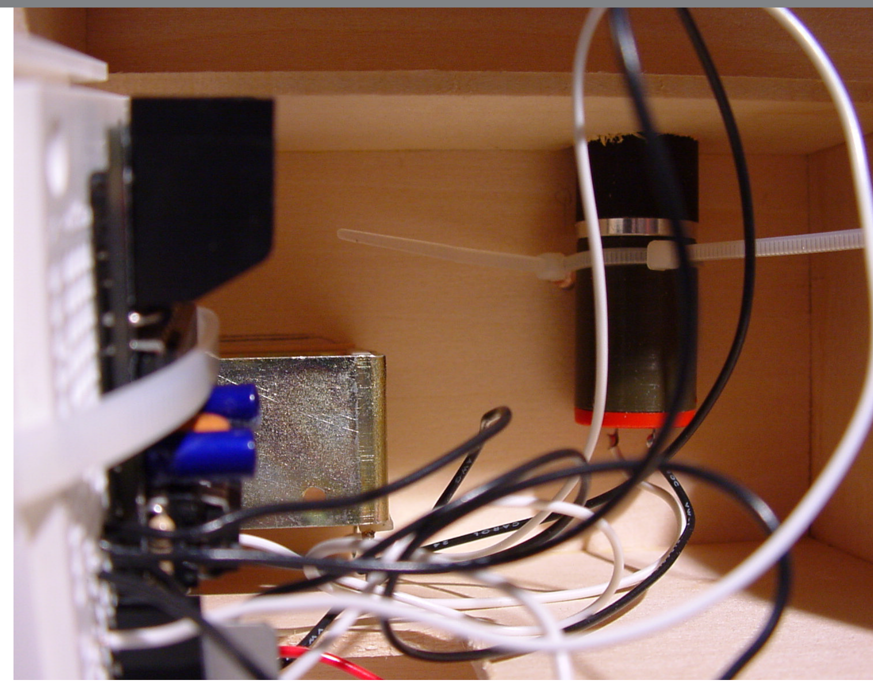
The ambient box moves in response to sound and light levels in its environment. A microphone attached to the front of the circular panel constantly monitors the sound level in the room and moves a motor attached to this panel quickly if there is a sudden change in the volume and slowly if the change is slight. The yellow rectangles on the bottom box are each attached to solar motors which release their energy as circular motion.

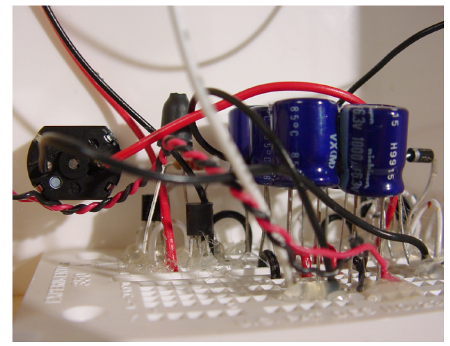












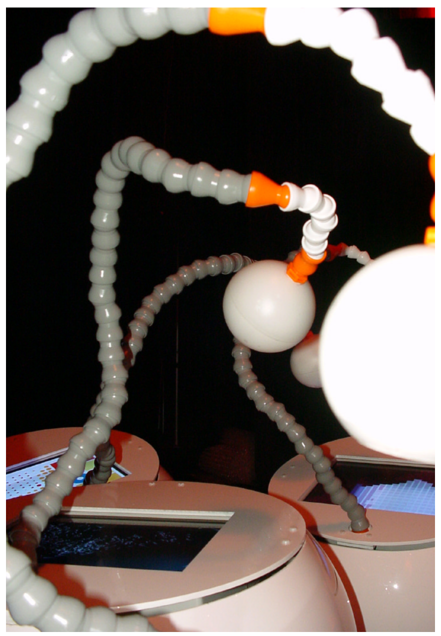
introspection machine

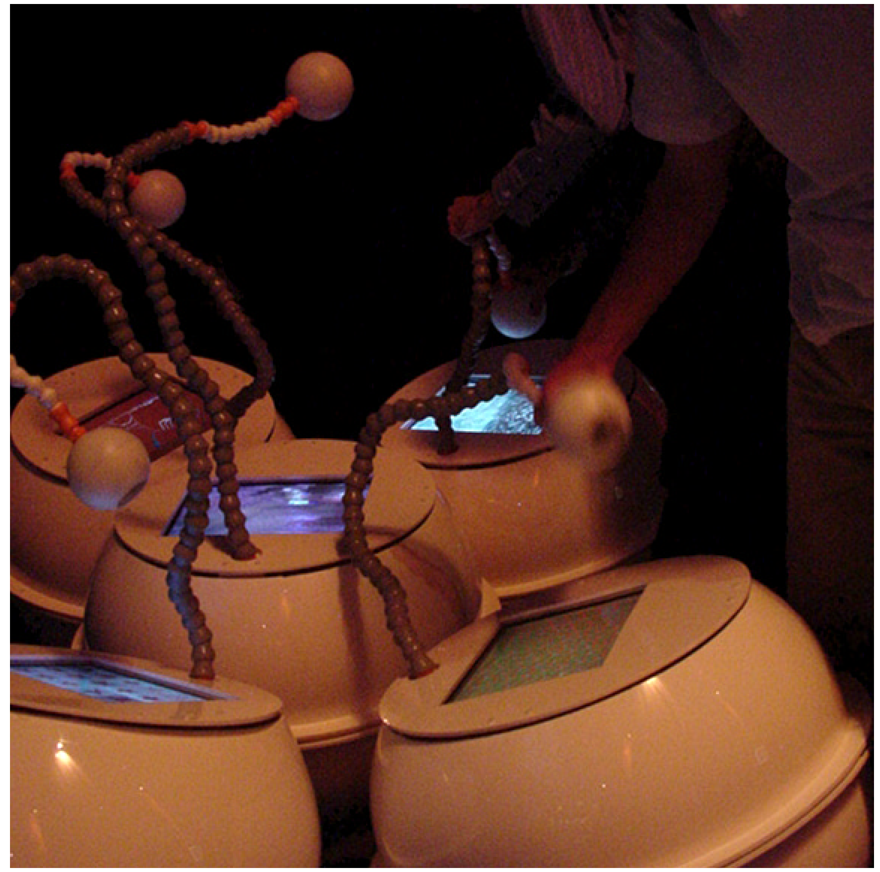
http://acg.media.mit.edu/people/creas/im

The Introspection Machine is an interactive visual feedback environment. The machine consists of multiple modules, each of which has a screen display and a flexible, manipulable eye. Each module transforms the video image from its eye into a dynamic display. By redirecting these eyestalks, users can explore an unbounded space of continuous light, complex forms, and surprising relationships.

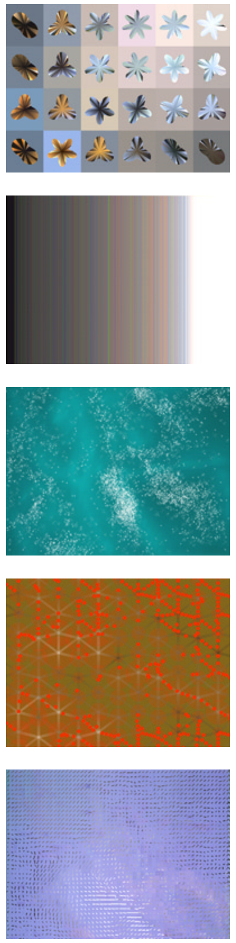
The reconfigurable eyes comprise the principal interface by which participants interact with the installation. Light is transferred from computer to computer, making it possible for the video output from one reactive display to be used as the input for another. An Introspection Machine module may also be piped back to itself, creating a tight loop of visual recursion. As visual material from each display is reinterpreted by the others, visual patterns shift and mutate based on the connections, configuration, and movement of the stalks. As a complex feedback system, the Introspection Machine has analogies to a visual network.

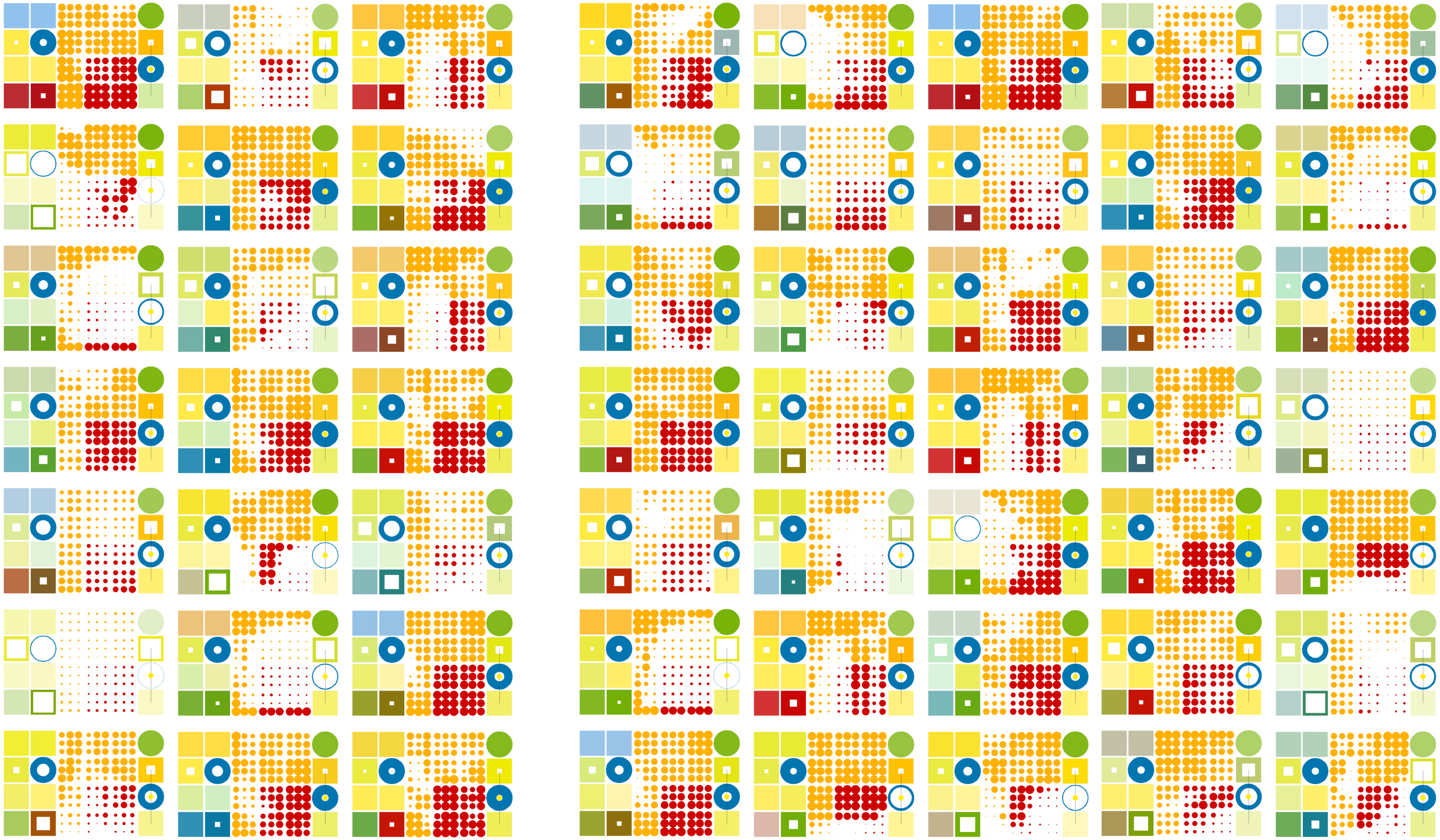
The Introspection Machine was created by the Aesthetics and Computation Group at the MIT Media Laboratory. It was first exhibited in the Art Gallery at Siggraph 2000.





Multiple software modules were written for the Introspection Machine hardware platform by the different members of the Aesthetics and Computation Group. From top to bottom: Booba by Elise Co, Disgrand by Ben Fry, Flurry by Golan Levin, Tri-way by Jared Schiffman, Springfield by Tom White. My module, Console, is shown below.

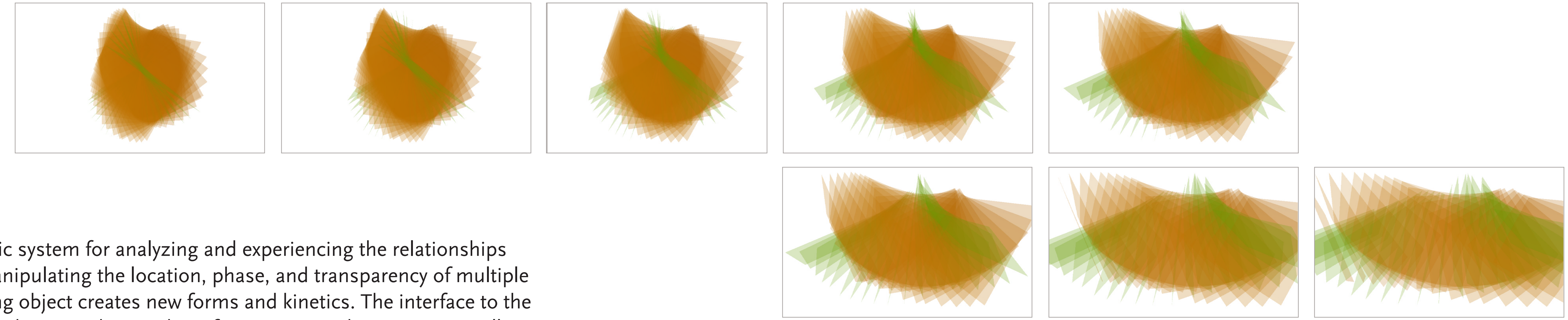




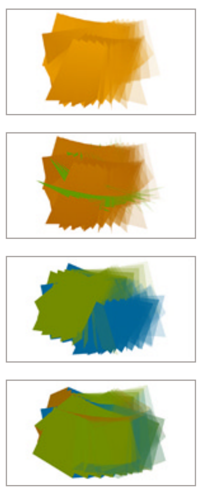
Each grid element acts as an aperture, opening and closing or changing their transparency in relation to the amount of light in the video input. Color is determined through a relational mapping to the original RGB color signal.

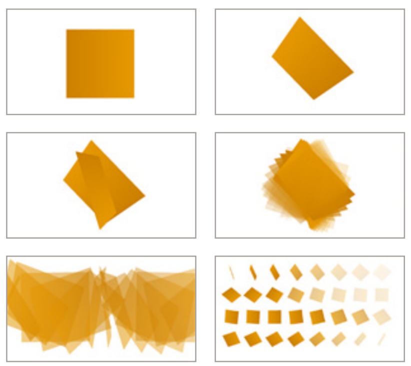
plane modulator

http://acg.media.mit.edu/people/creas/planemodulator

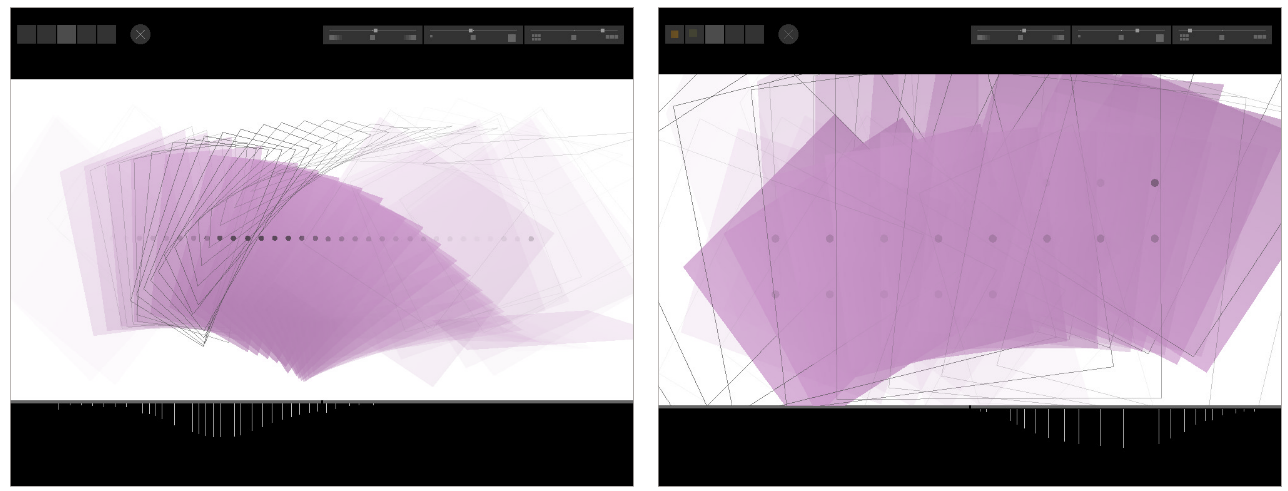


Plane Modulator is a dynamic system for analyzing and experiencing the relationships between time and space. Manipulating the location, phase, and transparency of multiple instances of the same moving object creates new forms and kinetics. The interface to the system provides controls for changing the number of instances on the screen, as well as their size, position, speed, transparency, and phase.



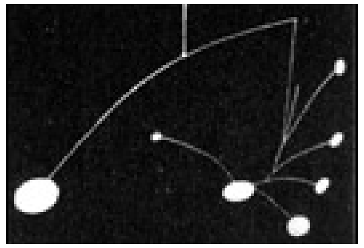


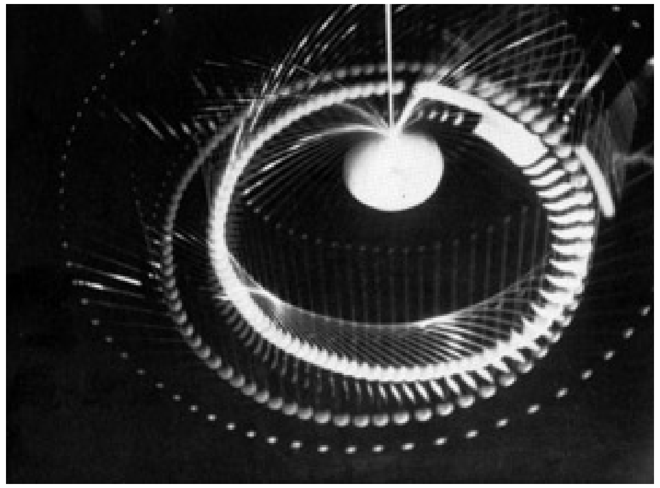
Rotating a simple plane about an arbitrary axis produces a dynamic quadrilateral. The addition of a second instance of the plane at a different rotation begins to construct the movement of the object through space. Through adding more instances, the virtual volume of the rotation reveals itself, but the shapes of the individual planes become obscured. Increasing the horizontal and vertical spacing between the planes clarifies their individual movements.



An interface was created to provide control over the geometry. The speed, size, and positional placement of the objects can be manipulated and a near limitless number of geometric primitives can be added to the system. Creating different patterns in time on the lower portion of the screen relate to patterns in space in the main image area.

Herbert Matter
Calder in Motion, 1939

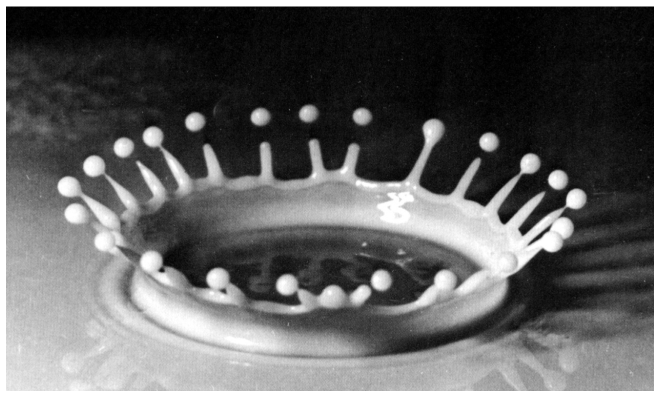


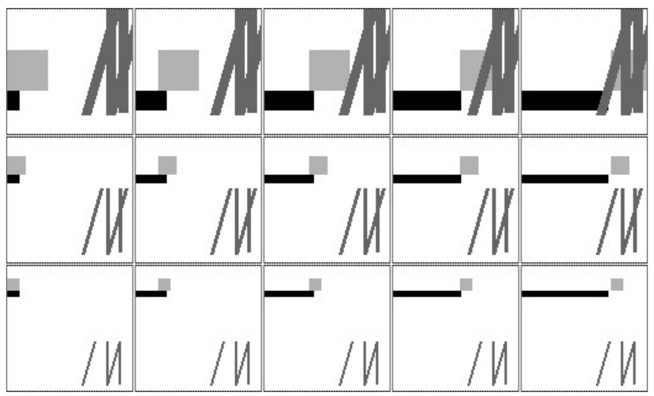


Plane Modulator is inspired by the history of technology used to augment vision. After the development of the camera in the mid 19th century, many artists began using its potential to stop time and analyze details of movement that had never been visible to the human eye. In the 1930s, Harold Edgerton pioneered the process of stroboscopic photography which enabled capturing the successive movements of an object with a still camera. Utilizing this technique, the photographs of Herbert Matter and L. Moholy-Nagy began to reveal a new vision, a sight unbound from time.

Digital computation has provided a medium for further extending our vision. Some of the experiments of John Maeda reveal this in an eloquent way. In his experimental Parametervision, every possible image constructed by a two dimensional parametric system can be viewed in a single image plane.

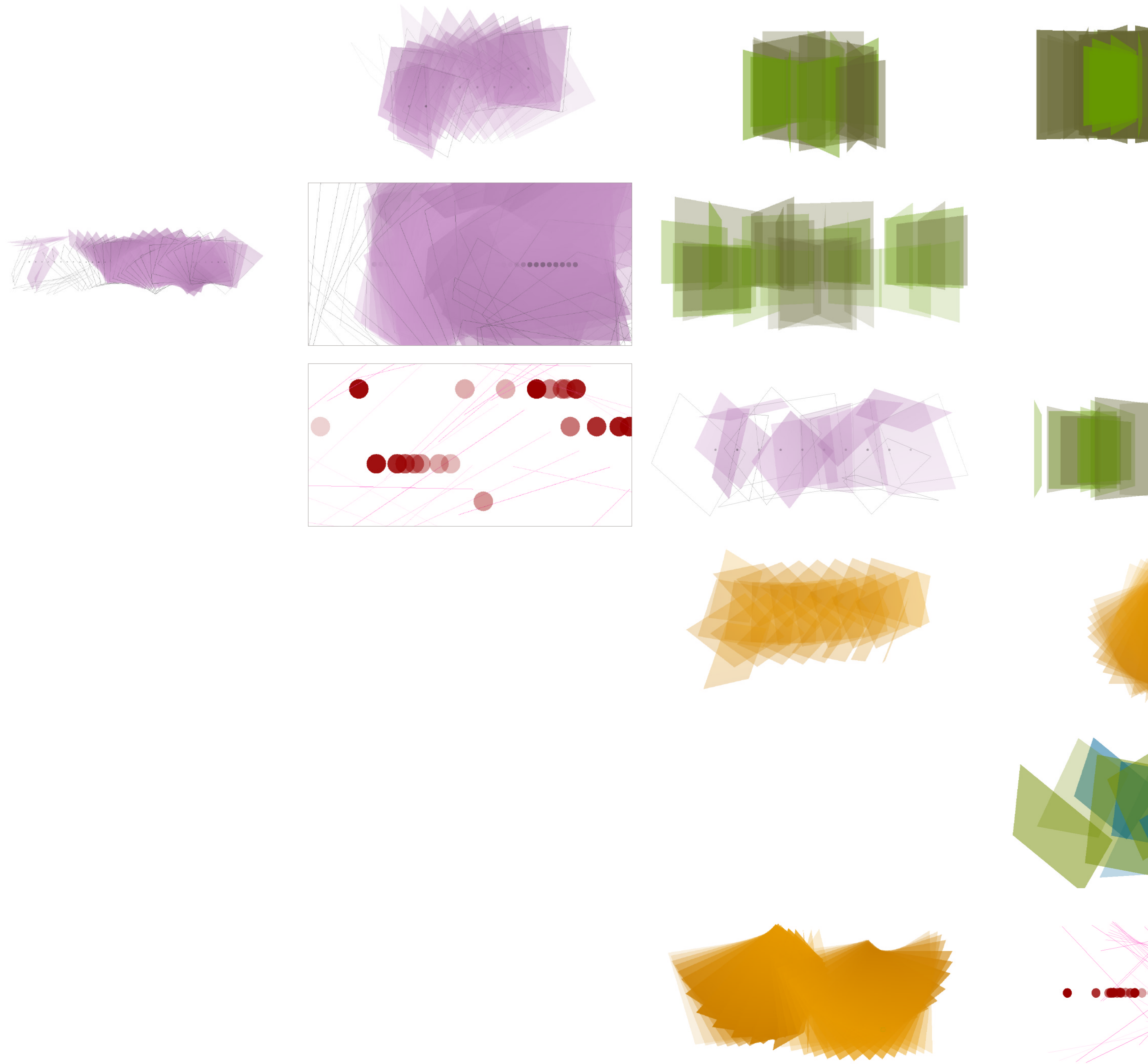
Plane Modulator combines both innovations into a system for constructing a new vision. It augments the early experiments in time and motion by providing the tools for analysis made possible through computation.

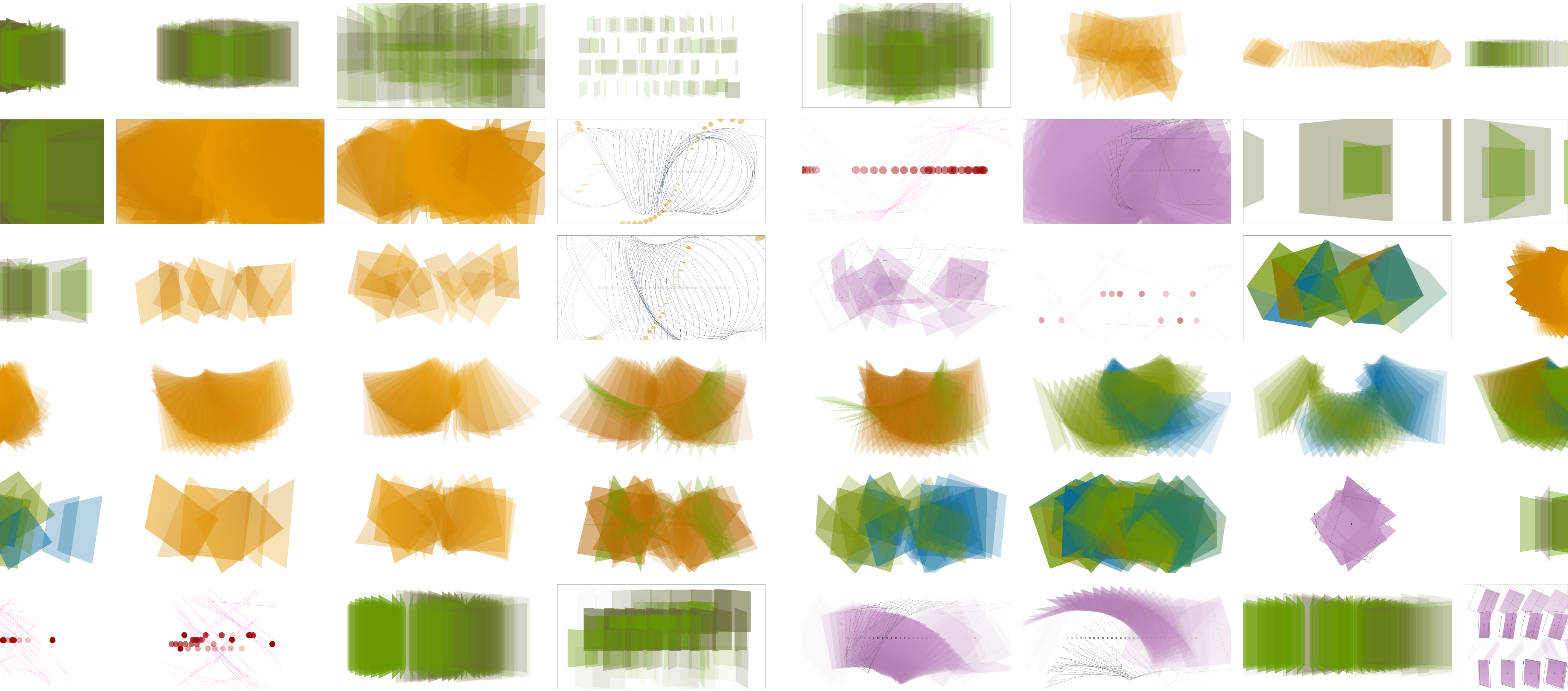


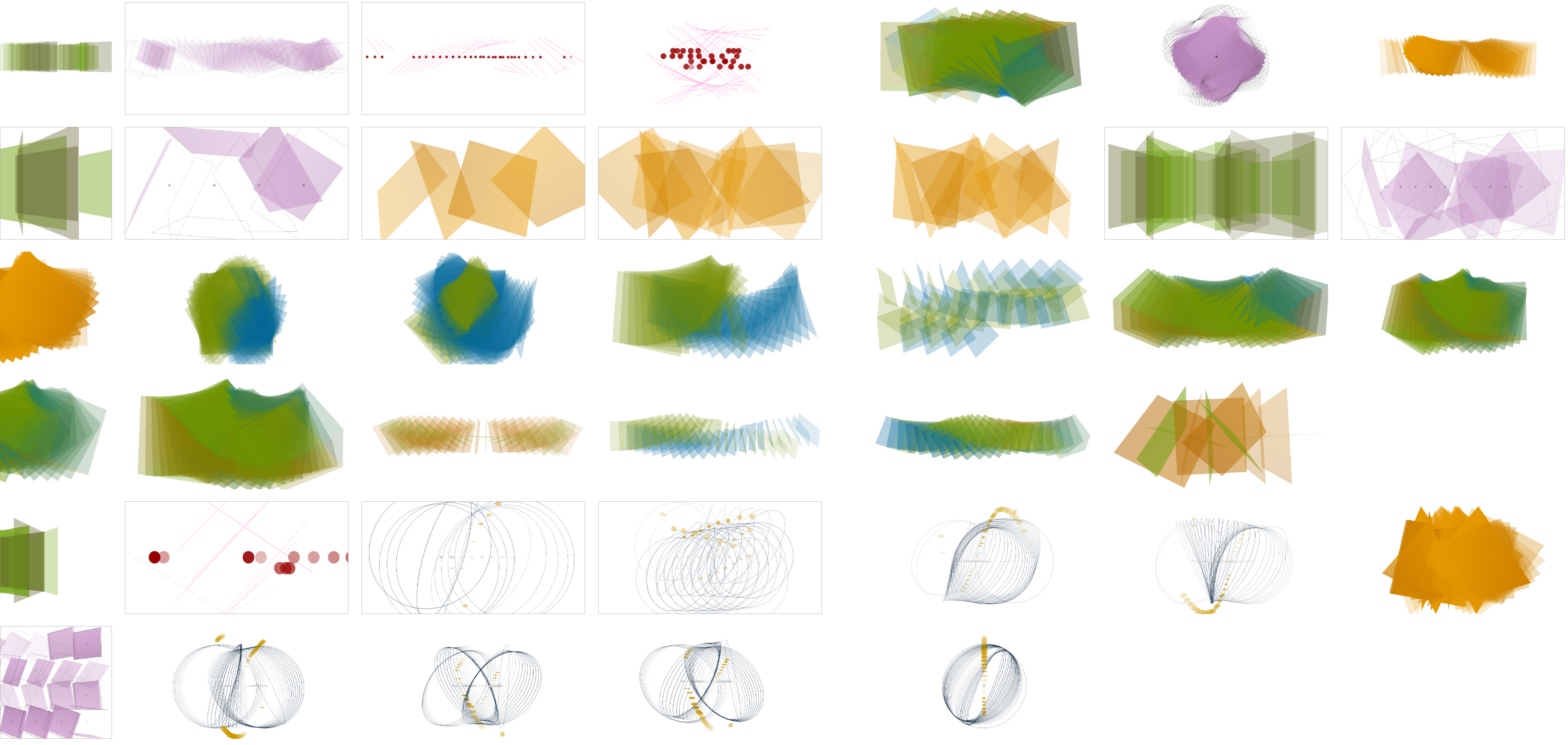


Harold Edgerton
Milk Splash

John Maeda
Parametervison, 1999



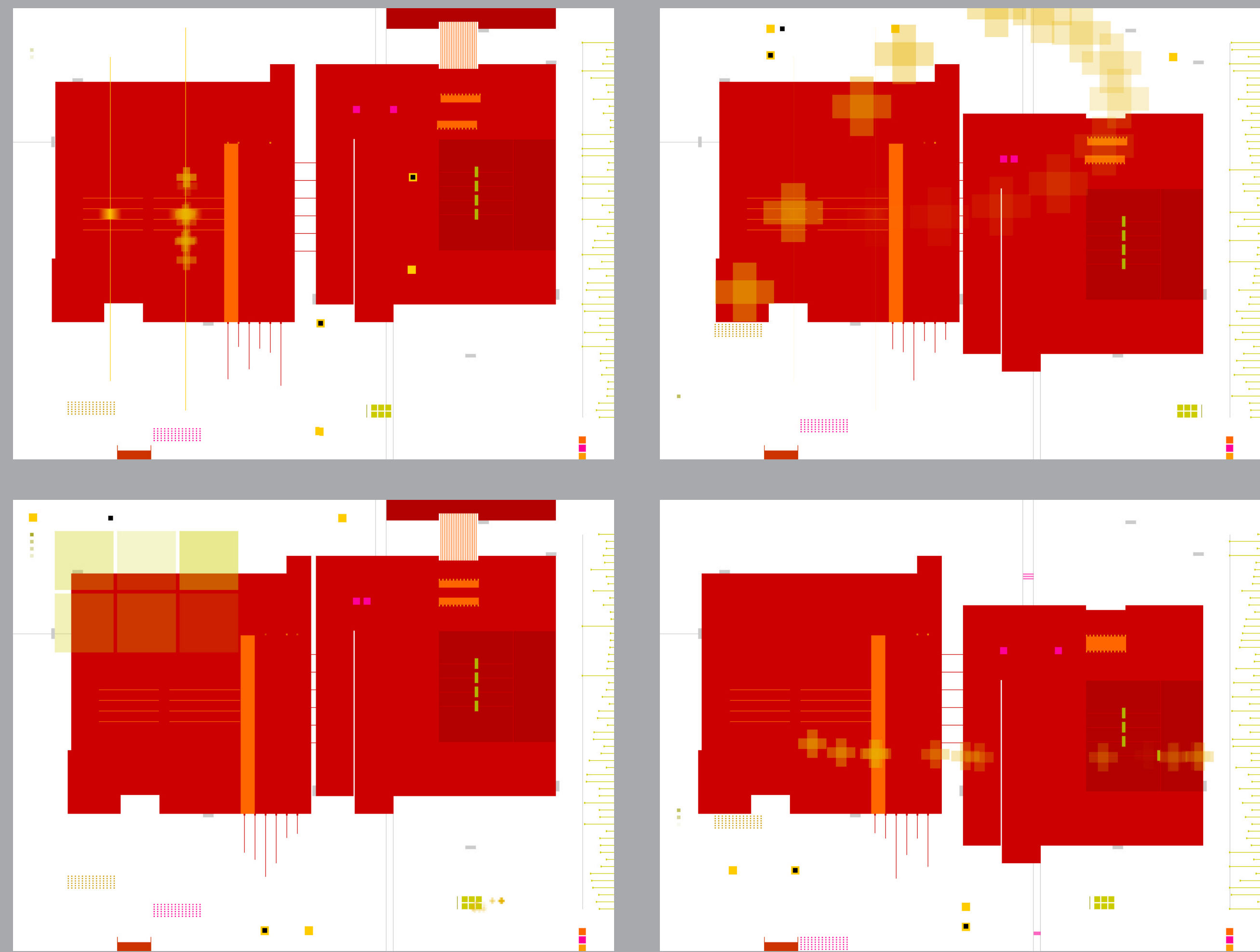




egg machine

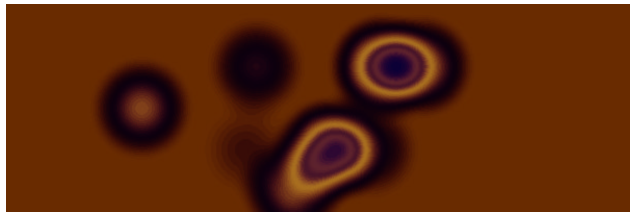
http://acg.media.mit.edu/people/creas/eggmachine

Egg Machine is an interactive audiovisual system. Intentional operation requires decoding the visual grammar of the system and manipulating the relationships between visual elements.

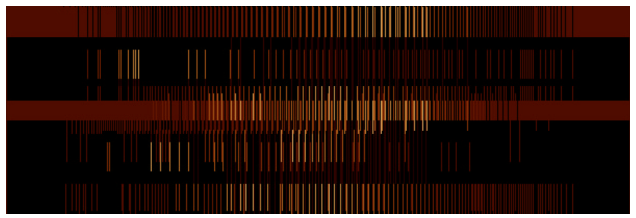


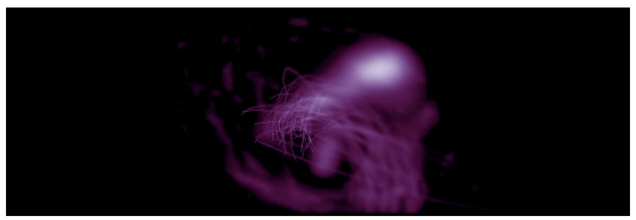
dakadaka

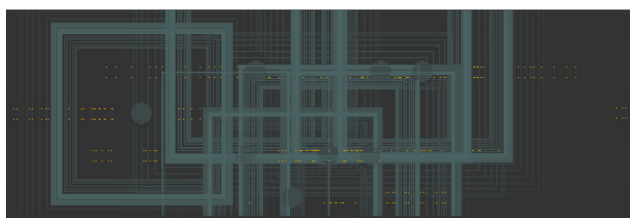
http://acg.media.mit.edu/projects/dakadaka

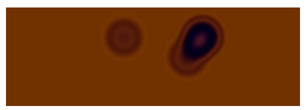


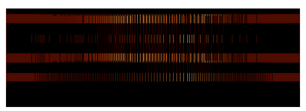
Typing can be thought of as a percussive spatial action—a play of tiny thoughts scattered onto a tightly organized grid. A kind of speech, spoken through the fingers with flashing rhythms and continuous gestures. Dakadaka is an interactive java applet that explores the processing of typing by combining positional typographic systems with an abstract dynamic display. Dakadaka was built in collaboration with Golan Levin.

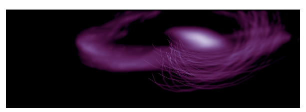


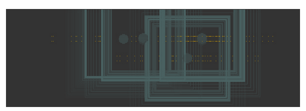


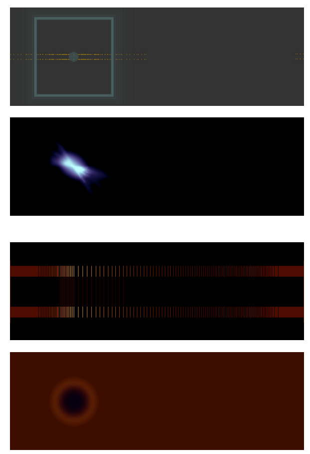










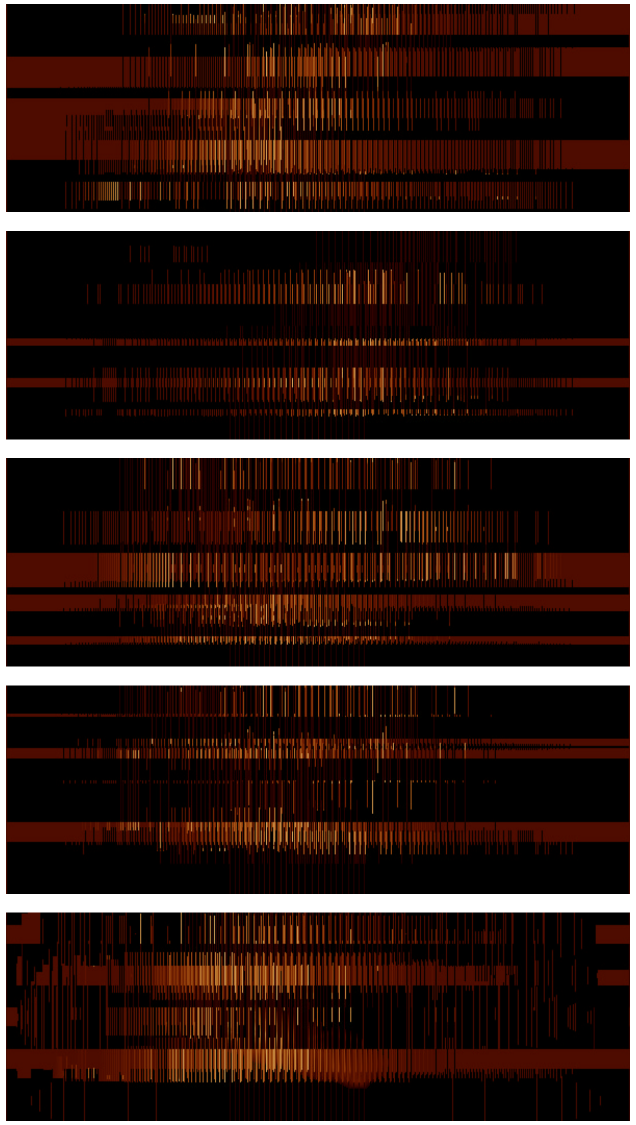


above
A networked version of Dakadaka was developed for the Ars Electronica Center's *Print on Screen* Exhibit in Linz, Austria. In this configuration, it is possible for one keyboard to conrol two or more computers at once.

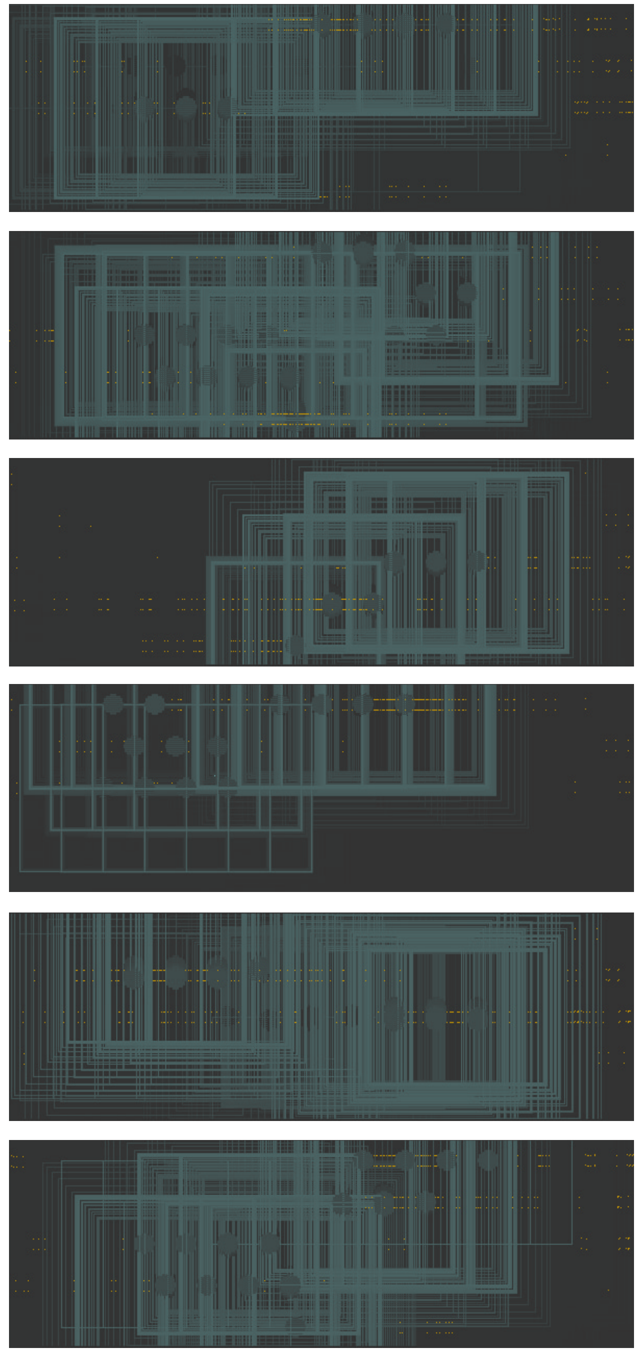
left
The foundation of Dakadaka is a positional typographic system. The X and Y coordinates of the keys on the keyboard are visually interpreted. In these images, the letter "s" was pressed on the keyboard.

worm_mode, Golan Levin

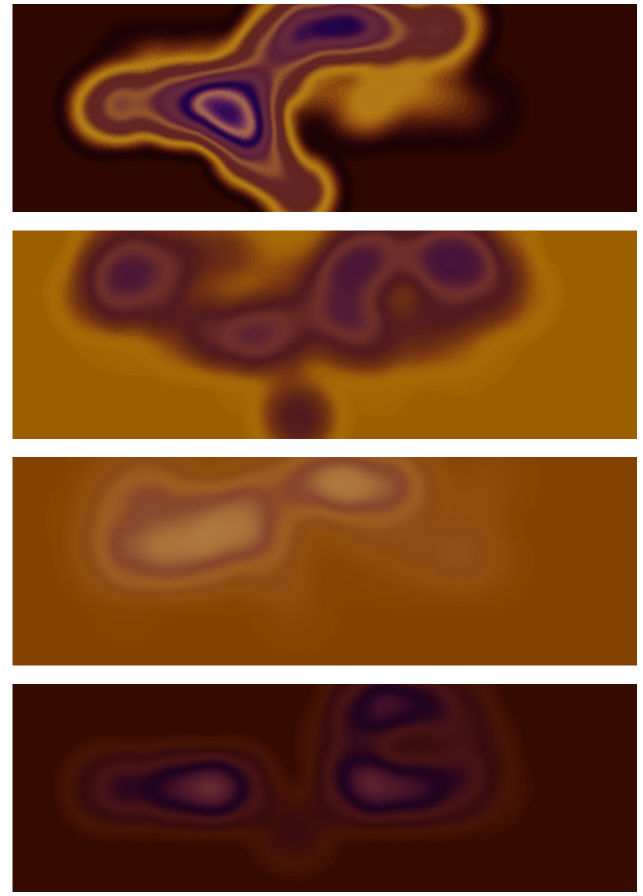
line_mode, Casey Reas



daka_mode, Casey Reas

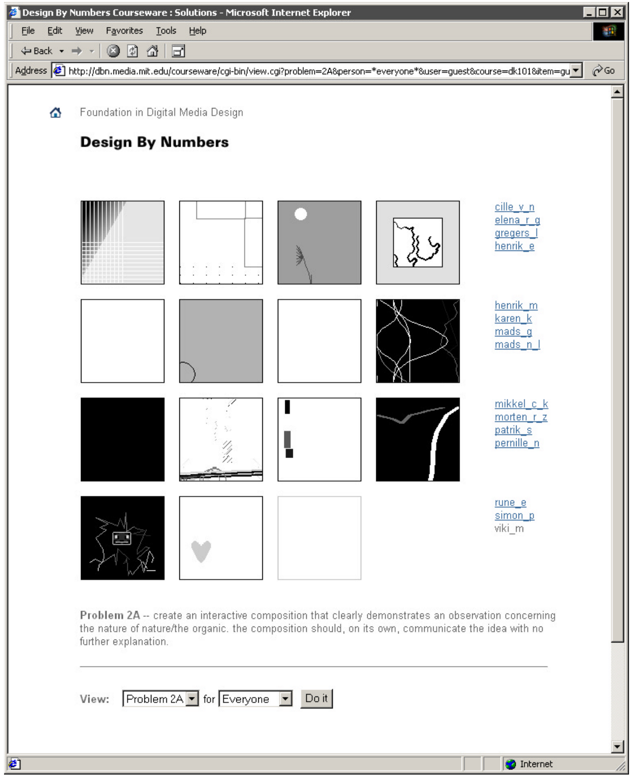


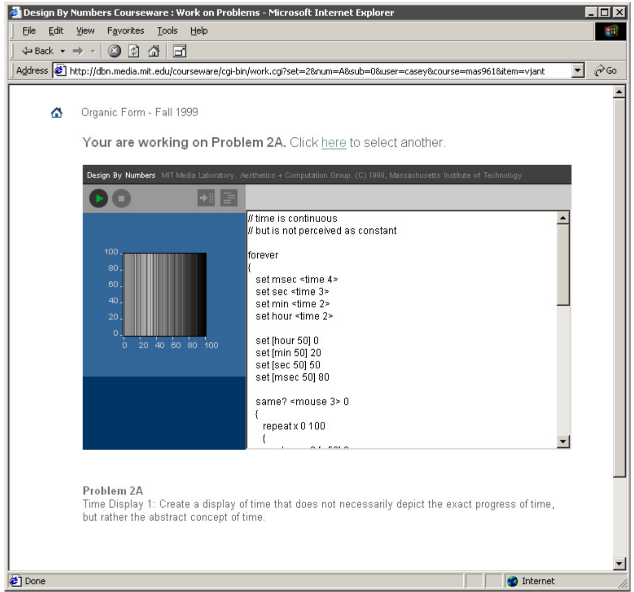
blob_mode, Golan Levin



design by numbers courseware

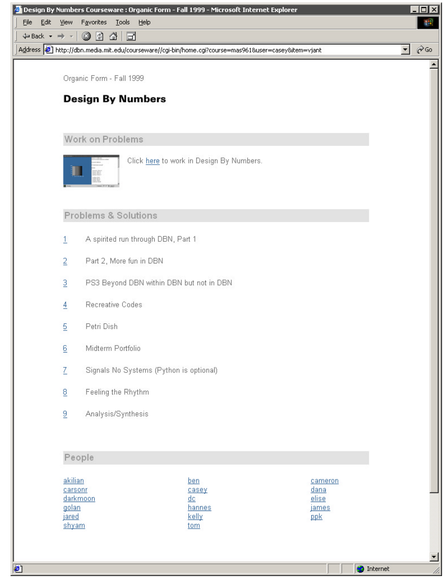
http://dbn.media.mit.edu

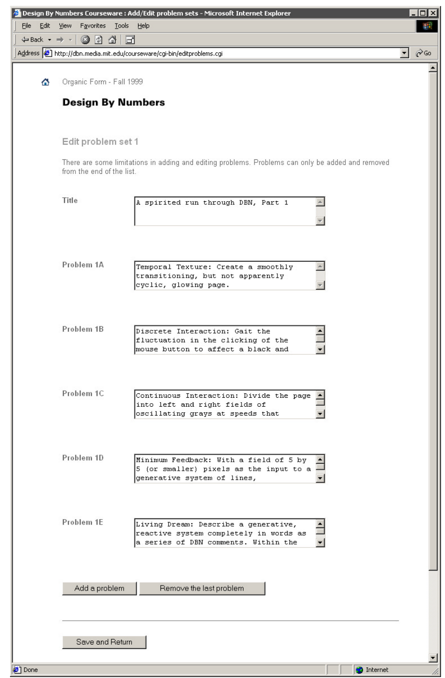




The Design By Numbers (DBN) Courseware is a web-based environment built to facilitate the use of DBN* as an educational tool for teaching computational design. It provides educators with an interface for inputting and editing problem statements and students with the ability to work in DBN and save their work directly to the server. Most importantly, it presents a unique viewing environment where students can critique their work within the context of the entire class.

* Design By Numbers is both a programming environment and language. The environment provides a unified space for writing and running programs and the language introduces the basic ideas of computer programming within the context of drawing. Visual elements such as dot, line, and field are combined with the computational ideas of variables and conditional statements to generate images.





Institutions using the Courseware include:
Designskolen Kolding—Kolding, Denmark
Massachusetts Institute of Technology—Cambridge, MA
Parsons School of Design—New York, NY
University of Texas—Austin, TX
The University of Tokyo—Tokyo, Japan
Tama Art University—Tokyo, Japan
Ravensbourne College of Design—Kent, United Kingdom
Columbia University—New York, NY

The Courseware has been used to teach seminars at:
AIGA National Headquarters—New York, NY
Fabrica—Novara, Italy
Icograda—Seoul, Korea
Rhode Island School of Design—Providence, RI

facing page, left
The Courseware provides a unified critique space where work can be evaluated individually or within the context of the rest of the class. The DBN code for the solutions is easily accessible from this page.

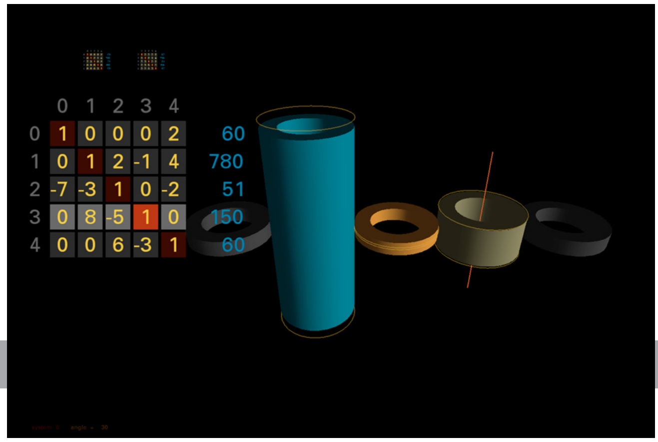
facing page, right
Students can work on solutions to the problems within the Courseware environment and save their work directly to the server.

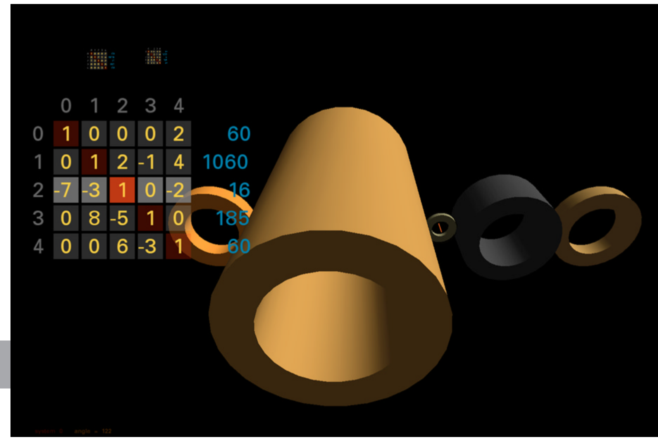
top
The Courseware includes a digital billboard for posting problem sets and links to viewing the work.

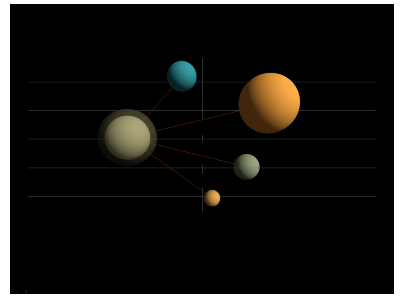
bottom
The course administrator can edit the problem statements and class roster at any time.

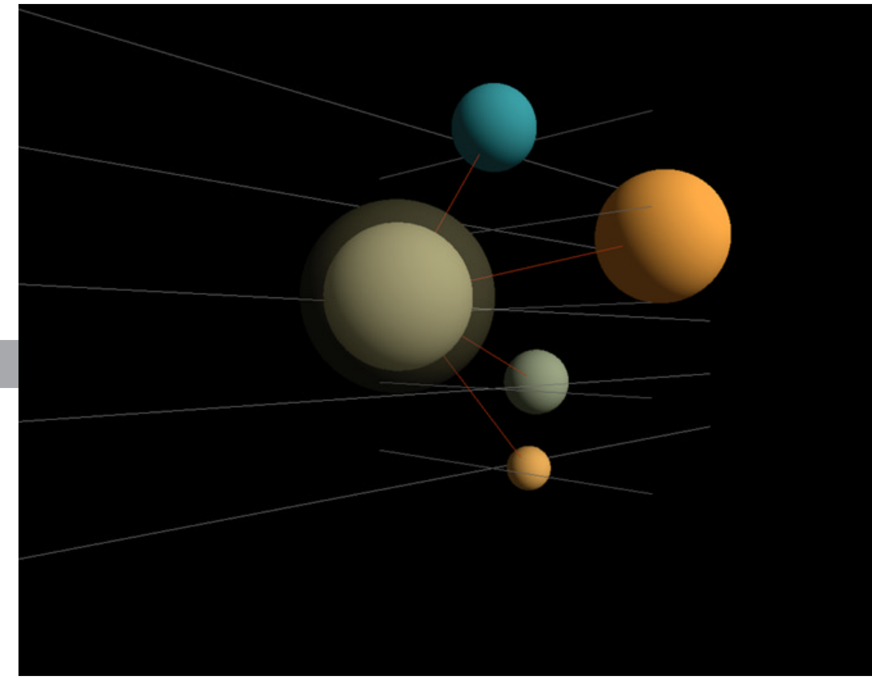
relational constructs

http://acg.media.mit.edu/people/creas/relationalconstructs







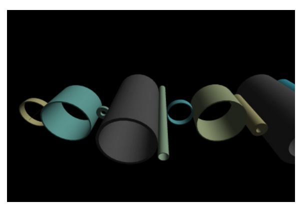


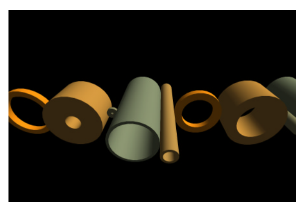
What are the aesthetic and experiential possibilities for dynamic compositions whose elements have an understanding of themselves and their relationship to the other elements? For example, what if the composition was homeostatic, constantly trying to maintain equilibrium regardless of external stimulus or if each element knew its visual properties and how to change them based on the current and previous data input into the system.

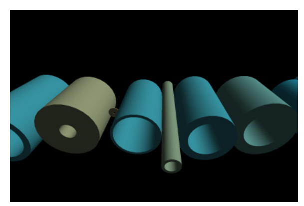
The Relational Constructs project focuses on the creation of such systems and visualizing the relations that define them. It is a tripartite system composed of raw data, a malleable set of relations which defines the system, and the result of the data being fed into the system (the final composition).

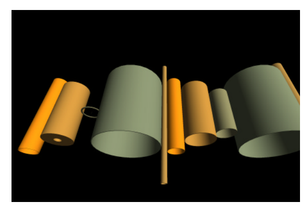
The project to date has worked toward visualizing the quantitative data elements of a system. The image above shows the current data values for five separate elements in a hypothetical composition. In this example, each element is represented as a sphere and has three values associated with it. These values are presented as the x-coordinate, z-coordinate, and the size of each sphere. The color communicates the potential energy for the increase of a value according to the system definition.

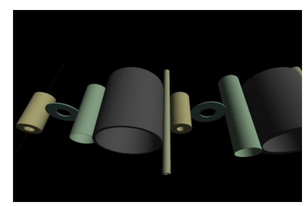
The spherical representation proved problematic so another form, the pipe, was chosen. Using the height, and interior/exterior radius values for the presentation of the data, the state of the system could be quickly and accurately determined. The addition of the matrix element was added as a holistic visualization of the system.





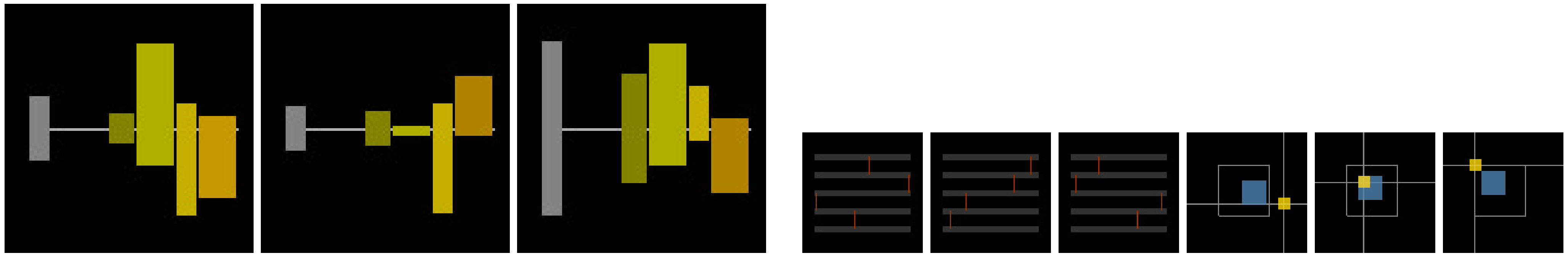






reactive 006

http://acg.media.mit.edu/people/creas/eat/reactive006



Reactive 006 is a collection of 32 studies exploring basic concepts in computational design. Each study expresses a single idea. Themes addressed are movement, visual relationship, interaction, recursion, flow, and space.