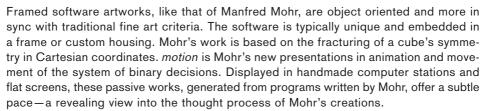# bitforms gallery

Steve Sacks

I started *bitforms* to explore the realms of digital art. To redefine categories and levels of artistic engagement. To discover new art. To educate both new and old school collectors.

bitforms represents artists who are using digital tools as an integral part of their process. These tools offer artists new ways of interpreting, manipulating and visualizing information. It's evolution in art. As a gallerist specializing in this type of work, it's important for me to know the tools. How and why are they being used? Are they necessary? Gratuitous use of technology for quick impact can be a detriment to both the artist and the gallery. The types of work we show include: Reactive sculpture. Data visualization. Sound and video installations. Digitally derived sculpture. Photo manipulation. Mixed media. Software art. Although there is a common theme of using digital tools, the idea and execution of the work are top priority. Is the idea fresh? Thought provoking? Relevant? Engaging? Granted, a lot of the answers are subjective, but there is an historic comparison and discovery process that can be used to define and position all works. In fact, artists have been using technology as an influence or tool for many centuries. But the use of digital technologies is different. Especially the use of software to create or to influence a creation. The category of art that has been most challenging for *bitforms* to define, market and legitimize has been software art. How do you collect it? Define it? What is its value? Future value? Archivability? Maintenance?

There are two categories of software art that *bitforms* represents—framed and unframed. The terms may sound traditional, but the issues involved are very different from framed and unframed prints and canvases. There are technical and use considerations in both types. Other considerations are networked, interactive, reactive and passive. These are terms that offer the collector different experiences and may require alternative maintenance and care.

Framed software art

Framed software artworks, like that of Manfred Mohr, are object oriented and more in sync with traditional fine art criteria. The software is typically unique and embedded in a frame or custom housing. Mohr's work is based on the fracturing of a cube's symmetry in Cartesian coordinates. *motion* is Mohr's new presentations in animation and movement of the system of binary decisions. Displayed in handmade computer stations and flat screens, these passive works, generated from programs written by Mohr, offer a subtle pace—a revealing view into the thought process of Mohr's creations.

In Daniel Rozin's *Mirrors #2, #5, #6,* he has created reactive screen-based software art pieces. They involve the viewer in a mirror type interaction. A video input interprets the form and each mirror delivers an expressionist compilation of color, shapes and movement. All three pieces deal with the way digital images and motion are reproduced and perceived.

Framed works are sold as variant editions or one of a kind and are typically much higher
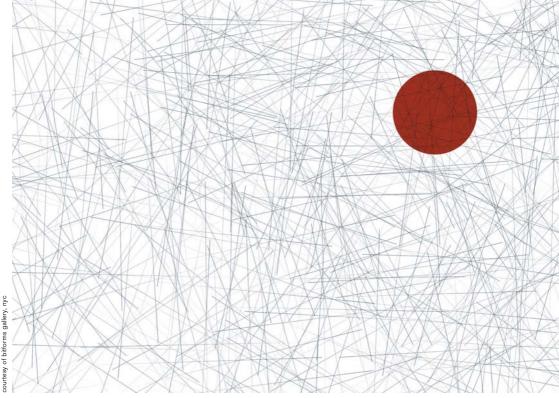
priced than unframed. With framed works there is complete control by the artist as to how the artwork will function and be displayed. Many times the framing or housing of the software is conceptually tied to the work. Some artists collaborate to fabricate their objects, where as others have the skill set to build the entire artwork. The software art objects need to be robust and easily maintained. Typically there is a guarantee and a detailed maintenance manual. Since there is hardware involved, care must be given to use.

Unframed software art

The unframed software art is a bit more difficult to define and control. It is sold on a CD and can be framed or displayed in any way the collector desires. The art can be interactive and passive. Networked and stand-alone. *bitforms* has sold stand-alone pieces in editions ranging from 10 to 250. Each CD is signed and in a custom package.

At *bitforms* we are adamant about how the work should be presented. An education process is necessary. The education starts with explaining how to present and engage the work. We recommend a dedicated machine and monitor to run the works (a software art station). At *bitforms* we have an ideal setup, two 18" touch screens floating in a rotating steel arm. Hidden is a Dell CPU with wireless network, mouse and keyboard. Presentation is important. To help people manage their software art, we have created an administrative tool that allows collectors to add, delete and select pieces. Once the software art is loaded and added to the system, the collector doesn't have to go back to the desktop. This is an important step in creating an isolated system that is focused on the viewing and interaction of software art. One system that we recommend for displaying software art is the all-in-one units from ezscreen. These are custom units that have
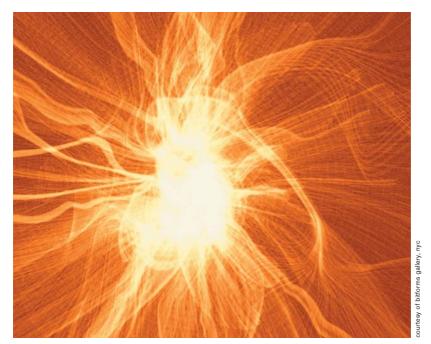
Casey Reas: *Tissue*

both CPU and touchscreen as one piece of hardware. They come in 15" and 18" versions, and hang on the wall like a painting or wall sculpture. The simplicity and flexibility of this system makes it very easy to collect and present most types of software art.

Collectors of this type of art vary. There are the old school collectors who are intrigued with the work and the low entry cost. There are the new collectors who are excited about the new technologies and the interactive nature of certain works. And there are museums that want to maintain a link to what is current and new. Some common questions collectors have are, What am I actually getting? What is the artist's role? Is this really art? The collector is getting a set of rules or parameters determined by the artist. These rules are the art. They are the essence of the experience and aesthetic direction. The rules can be interpreted as the code the artist writes. The questioning of this as an art form is to be expected.

One of the issues with unframed software art is this concept of display on a dedicated system. Even though the cost of a flat screen and CPU have come down drastically over the years, it is still a psychological hurdle to dedicate a computer to one task. So, many people choose to buy the work and place it on their everyday machine with all of their stuff. I equate this to buying a print and putting in a magazine. It is a distraction and takes away from the experience. The other hurdle is the "screen saver" comparison. Society has chosen to consider screen savers with very little regard—they are temporary visuals. Another challenge for the legitimacy of this type of art.
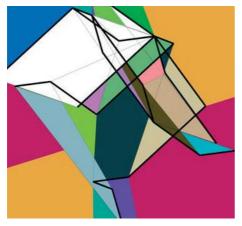
Another type of unframed software art uses a network or Internet connection. One example is Mark Napier's *Waiting Room*—a collaborative network piece within a virtual space that 50 users share through the Internet. In this space the visitor becomes a participant in a moving painting. Their actions activate and shape the artwork, shifting the screen
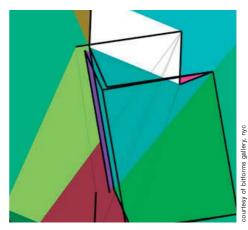
Golan Levin: *Floccus*

through moods, from hard-edged to atmospheric, from dark to light, from quiet to chaotic. Each click creates a shape, a shadow or a wall, a suggestion of architecture or a dissolving light. Since there is only one piece that exists on a server, we sold this work in shares. 50 shares at $1,000 per share. A grass roots way of promoting this work was through a software art party at a collector's home. We had the *Waiting Room* running at the party. The collectors who couldn't attend were connected to the piece and interacting with those at the party. Mark Napier was there to discuss the piece and answer any questions. This was an ideal way to experience the true beauty of the collaborative network art concept in a social setting. It also let people see how to live with the art in a variety of ways.

Works from Golan Levin and Casey Reas have ties to abstract expressionism. Their works are organic interpretations of form, motion and interactivity. In *Floccus,* by Golan Levin, ductile filaments drawn by the user swirl around a shifting, imaginary drain centered at the user's cursor. These filaments—torn by conflicting impulses to simultaneously preserve their length, yet also move towards or away from the user's cursor—find an equilibrium by forming gnarly, tangled masses. *Tissue,* by Casey Reas, exposes the movements of thousands of synthetic neural systems. Each line in the image reveals the history of one system's movement. People interact with the software by positioning a group of points on the screen. By positioning and re-positioning the points, an understanding of the total system emerges from the subtle relations between the positional input and the rich visual output.
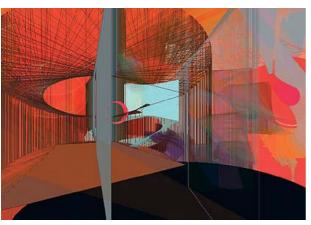
Software art is empowering. Engaging. Endless. Whether or not it becomes a valuable collectable, I am convinced that it will be a part of the art nomenclature. Its beauty and possibilities are too alluring. The artists are too talented. And the world deserves a new creative outlet.
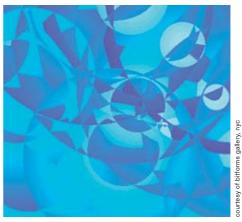
Manfred Mohr: *P-707/C11, 2001*
Endura Chrome, canvas, vinyl

Manfred Mohr: *P-708/C, 2001*
Medium: Endura Chrome, canvas, vinyl

Mark Napier: *Waiting Room,*
2002 System Specs
CD w/interactive software art

Mark Napier: *Pea Soup, 2000*
Java applet, C code
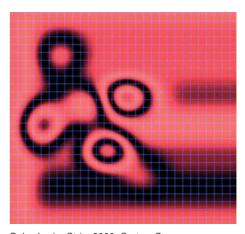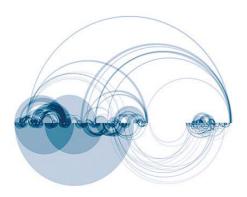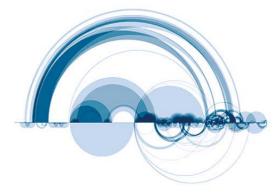
Golan Levin

Golan Levin

Mark Napier

Golan Levin: *Transmission (Undeciphered),*
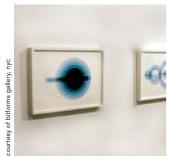*2001 System Specs*
CD w/interactive software art

Golan Levin: *Stria, 2002. System Specs*
Medium: CD w/interactive software art

Martin Wattenberg:
Stairway to Heaven (Zeppelin), 2002
Medium: Archival digital print, signed by artist

Martin Wattenberg: *Cocaine (Clapton), 2002*
Archival digital print, signed by artist

Martin Wattenberg

Martin Wattenberg

Daniel Rozin