About... Software, Surveillance, Scariness, Subjectivity (and SVEN)

Amy Alexander

Department of Visual Arts,
University of California, San Diego
9500 Gilman Dr. #0084
La Jolla, CA 92093-0084 USA

Abstract. The text discusses cultural and political implications of the subjective aspects of software and the SVEN project.[1] SVEN (Surveillance Video Entertainment Network) is a public space software art project that uses custom computer vision software to detect pedestrians who in some way *look like* rock stars. The text introduces general audiences to SVEN's approach to software subjectivity—in this case, concerning computer vision surveillance software. It also presents examples of software bias in contemporary culture and proposes software literacy as a public educational goal.

Keywords: surveillance, software art, digital street performance, sousveillance, computer vision, software literacy.

1 Introduction

SVEN (Surveillance Video Entertainment Network) is a project developed by Amy Alexander, Wojciech Kosma, and Vincent Rabaud with Jesse Gilbert, Nikhil Rasiwasia, and Marilia Maschion. The following text focuses on SVEN's approach to and issues surrounding computer vision. Cinematography, and its relationship to both software and surveillance video, is also important to SVEN, but it's a topic for a different text. (Art is of course of particular importance to SVEN—but that should go without saying.)

SVEN is a piece of tactical software art. Tactical software art comes out of traditions of tactical media and software art. It's a logical mix: tactical media is a response to the way mainstream media influences culture; software art is a response to the ways mainstream software influences culture. Tactical media often involves a combination of digital actions and meatspace—or street—actions. In SVEN, these are one and the same—digital actions that take place on the street (just off the curb in this case).

SVEN is a self-contained computer vision-based surveillance system that is designed to detect likely "rock stars." In its street performance context, the system is

[1] More info on SVEN can be found on its website, http://deprogramming.us/sven

R. Adams, S. Gibson, and S. Müller Arisona (Eds.): DAW/IF 2006/2007, CCIS 7, pp. 467–475, 2008.

installed in a cargo van that parks along a sidewalk with pedestrian traffic.[2] A video camera is mounted on the roof of the van, pointed at pedestrians half a block or more away. A video monitor sits in the van window displaying what appears to be surveillance video. As pedestrians pass through the camera's view, SVEN matches certain of their physical characteristics against those of rock stars as they appear in selected music videos. The selection of characteristics to match range from facial expression to clothing and hair color to body position. These characteristics are deliberately unorthodox, as though the surveillance system had become bored and longed to catch a person of interest and so grasped at any excuse it could for a match—algorithmic wishful thinking. Thus, when SVEN detects what it believes is a rock star, the normally boring surveillance video on SVEN's monitor erupts into a music video (with corresponding audio). This music video is generated in real time from the live video and stars the unsuspecting pedestrian. Along with the music video, the monitor displays two smaller images illustrating the match between the pedestrian and the rock star. The multi-view effect is similar to the arrangement of large and small monitors in a CCTV control room, where a large monitor shows the main view on which the staff are to focus their attention, and smaller monitors keep track of the activity taking place in front of individual cameras.

Fig. 1. SVEN in its van configuration—aka *SVAN*. A camera with telephoto lens is mounted on the front of the SVAN roof and points at unsuspecting pedestrians up to 150 meters away. The monitor in the window displays video to other pedestrians as they pass the van. Speakers just inside the van window play corresponding music when the system is generating a music video.

[2] Besides its van-based street performances, SVEN has also been installed in storefronts and in public areas of a museum. While it's more common to concern ourselves with surveillance on the street, surveillance inside public places can be just as insidious.

Fig. 2. A passerby encounters SVEN in Zürich. The camera on top of the van is pointed at pedestrians approximately 100 meters down the street.

Fig. 3. Screenshot of SVEN display in music video mode. On the lower left is the live camera image showing the pedestrian tracked and her body segmented into head, head and shoulders, shirt, and legs. This segmentation data is used by the system both for matching characteristics of rock stars and for positioning effects and cinematographic framing (close-ups, etc.) On the upper left is the actual rock star and music video matched. The large frame on the right shows the result of SVEN's self-deluding algorithms: the live scene is transformed in real-time into the matched music video, featuring the tracked pedestrian as the rock star.

2 Surveillance Is Already Scary

Sure, surveillance is scary—but you've probably heard that before. We're being watched all the time, and we don't know by whom, or what they're doing with the images and other data they're gathering. Scared? You bet—there's a bogeyman under the bed, so we'd better not look. But remember, we're supposed to be scared—people are trying to scare us. Foucault pointed out that not knowing when the bogeyman is watching you can scare you into changing your behavior. But not knowing *how* the bogeyman is watching you can scare you too. SVEN's purpose is not to point out that surveillance is scary. People are scared enough as it is.

3 Software Shouldn't Be Scary

Technology functioning as a big "black box" often scares people into not looking at it. It's all-powerful and incomprehensible. So, people often don't question how it works. Although the significance of this state of affairs is often-overlooked, it's by no means a recent development. In 1987, William Bowles wrote about the risk of the loss of transparency from the likes of the Macintosh computer:

> … many people have raised serious objections to the "black box" approach used by machines such as the Macintosh, arguing that by making the machine into a closed system it not only reduces the range of choices open to the user, but perhaps more importantly it encourages a particular attitude towards machines in general by mystifying the processes involved, which in turn leads to a state of unquestioning acceptance of the supremacy of technology. This is of course a process which began with the industrial revolution. [1]

A more recent example—an article from WikiWikiWeb entitled "Hermetically Sealed Stuff Is Magic"—reads:

> This is a principle of human nature pointed out to me by ScottAdams and his PointyHairedBoss. There is a Dilbert strip where the PointyHairedBoss works out a schedule for Dilbert, and bases it on the assumption that anything he cannot understand is easy (magic). Thus, he commands the poor drone to build a worldwide networking system in six minutes.
>
> If you can understand something, you can reasonably evaluate it. If you can't understand it (either it is beyond your comprehension, or someone has "hermetically sealed" it so you can't see), you can't reasonably evaluate it. [2]

That might sound at first like a geek-elitist position, implying that everyone should be a programmer and that those who don't program are (lazy/stupid/inferior). I can't speak for the authors of that wiki article, but my point here is not to suggest that everyone learn to program, but rather that perhaps everyone should learn *about* programming: software literacy. Think of software literacy as an extension of media literacy. People are (hopefully) taught how to detect bias in newspapers and television—even if they don't know how to produce a newspaper or television program themselves. Now that software is a mass medium—one that influences people's lives at both consumer and institutional levels—might not it be useful if people learned to detect software's biases?

4 How Is Software Subjective?

Some real world examples may be useful in illustrating the subjectivity of software. Example 1: Google, whose search results significantly influence the information people access, touts the objectivity of their PageRank technology:

> PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important. [3]"

I'd argue that the algorithm described isn't "democratic" but is actually rather similar to becoming popular in high school. If the popular kids like you then you can easily become popular. But what if you're not part of the in-crowd? What if you're a dissenter—or just not trendy? According to the algorithm described above, it's difficult to get noticed. Google apparently refines the PageRank algorithm on a regular basis, and they keep its exact workings a secret. (If they didn't, it's likely we'd all see even more ads than we do for products that begin with a "V" and end with an "a.") But at least we can begin to critically question how PageRank influences the information we read. And even though Google assures us that "Google's complex, automated methods make human tampering with our results extremely difficult," (Google) we can keep in mind that humans determined the automated methods in the first place.

Example 2: The United States Internal Revenue Service was recently criticized for freezing the tax refunds of many poor taxpayers by targeting their returns as likely to be fraudulent—even though most were not. An article in *The New York Times* reported that "a computer program selected the returns as part of the questionable refund program run by the criminal investigation division of the Internal Revenue Service. [4]"

The article doesn't tell us any more than that about the computer program, but obviously someone programmed it with rules for finding a "questionable" return. Clearly, those rules were subjective, and they seem suspiciously like they may have been politically motivated. The fact that the deed itself was done by computer doesn't make the decision mechanical, blind or objective. In a software-literate culture, the journalist who wrote the article might be expected to press for details on how the program worked, or at least discuss his inability to obtain this information from his sources. But at present, it seems largely culturally acceptable to shrug such things off: "The computer did it."

5 On Algorithms and Data; Verbs and Nouns; Parts and Wholes

Of course, algorithms can't operate without data. A simplistic analogy for thinking about a software process would be to say that data are nouns and algorithms are verbs. I've discussed algorithms above—so, what about data? The idea that we live in a "database culture" is a familiar one: from playing computer games to shopping to going to the doctor, we face one database after another in our daily lives. And just as

we worry about our physical bodies being subject to visual surveillance, we also worry that our data bodies are subject to virtual surveillance. Are "they" watching my search habits, my browsing habits, my online purchasing habits? Naturally, we want to know *what* information about us is being used (nouns). But again, we need to look also at what *actions* (verbs) are being performed on or with the data. Consider the situation of registering for an account on a social networking site such as MySpace or Facebook. One is inevitably asked for age, gender, marital status, and various more personal questions. One may or may not find these questions individually inappropriate or prying, but what's less obvious is how one may feel about the ways in which the responses to these questions are put together. As Aileen Derieg wrote on the Furtherfield Blog:

> In the end, I found myself defined—seemingly voluntarily—over and over as a female over 40, married with children. By itself, this information is wholly devoid of any content, although it might well serve as a surface for myriad projections. Some anonymous stranger might read that as a description: traditional, conventional, conservative, maybe interested in cooking and gardening and parenting issues ... Or it might suggest a bored housewife potentially up for all kinds of illicit naughtiness, following a well established narrative from spam. As entirely inane and irrelevant as this is, however, what concerns me is how my goal of exploring possibilities of exchange and connections within the framework of "terms of use" and "privacy policies" defined by the respective corporate owners was initially deflected from the start through the rigid constraints of constructing an identity through the process of "registration. [5]"

As with many things, the whole can be quite different than the sum of its parts. And seemingly benign, objective computer algorithms such as the display of fields from a database can turn into something quite different when combined with such subjective human "algorithms" as interpretation.[3]

6 I'm Not Myself Today…

If we say someone "matches" a terrorist (or anything else)—what does it really mean? Some characteristics of that person's appearance have been determined to be significant—they match some terrorist's photo more closely than others in the database. This raises the question – what are these "significant" characteristics?

In "Face Recognition Using Eigenfaces,"[4] images document the results of researchers' attempt to use computer vision algorithms to match photographs of individuals with those in a database. The second grid of photographs from the top of the web page shows the results of an attempt to use computer vision algorithms to match black and white photos of test subjects with photos of the same subjects in a database. We see that the algorithm detected the correct person from the database in a large

[3] As of this writing (December 2007), semantic web technologies—those that allow objects on the web to be located by combining arbitrary user-defined criteria—are still in the early stages. Assuming these evolve and become widely used, I suspect there will be a lot more discussion on this topic.

[4] Accessible online at http://www.cs.princeton.edu/~cdecoro/eigenfaces

percentage of cases. [6] However, the few incorrect cases are interesting. The software attempted to detect similarity between photographs and faces—and it did so—according to *some* characteristics. Not, in these cases, the characteristics that would have given the "right" answer and identified the same person. But the wrong answers may not be what we expected – or feared. Instead of confusing people of the same race, for example, the software will sometimes confuse two people with a smug expression on their face. Maybe in some ways smug people have more in common than people of the same race. Maybe, on days when you're not yourself, you're really more like someone else. In any case, attitude profiling may turn out to be a greater risk of technology than racial profiling.

But profiling concerns aren't limited to race. If the computer vision bogeyman were used to identify "undesirables," what would those undesirables look like? Presumably, everyone could envision their own profile of an "undesirable." And in fact, such profiling could be programmed into a computer vision system. But—the profiles would need to be quantified for the computer. It turns out, computers are subject to the same sorts of stereotyping as humans are – only more so. For example, say you're on the lookout for troublemaking emo kids. You could tell a human, "Watch out for emo[5] kids," and this would be asking the human to stereotype. But you'd have to tell the software, "Detect people wearing all black, with pale skin and very black hair." This is more extreme stereotyping than the human might do, at least consciously. But of course, humans chose those characteristics.

So—one of SVEN's aims is to reflect on the human subjectivity inherent in technology. Because this subjectivity must be reduced to objective rules, such implementations obviously have limitations in mimicking the way humans would perform the

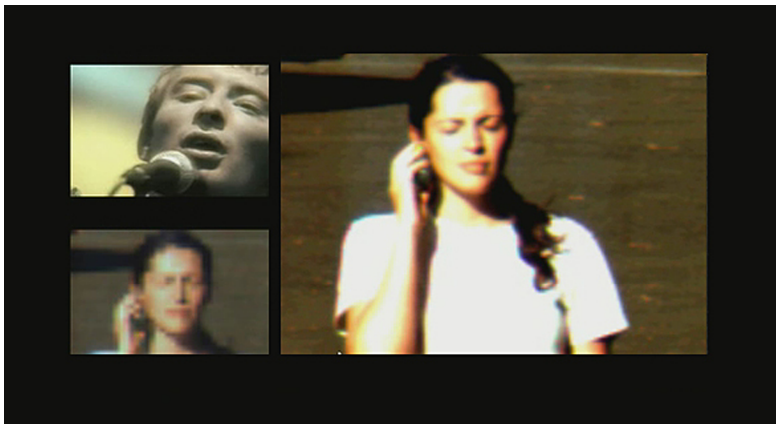

Fig. 4. Screenshot of SVEN detecting the resemblance between a contemplative pedestrian and Thom Yorke of Radiohead

[5] The term "emo" is used here to describe the stereotypical appearance of teenagers who have adopted the so-called emo fashion. This fashion involves, among other things, dyed black hair, and skin that appears pale (perhaps in contrast to their artificially darkened hair.)

intended task. However, the implementations and their results can, through these limitations and exaggerations, reveal less obvious things about how their human creators "see" things—and about humans in general. Technological development expects machines to think like humans and humans to think like machines – under this stress both give something about themselves away.

7 Technology and the Way It's Used Aren't the Same Thing

This might seem an obvious point, but the opportunities it presents for tactical software might easily be overlooked. Take for example, computer vision surveillance technology. It conjures up depressing connotations, and our gut reaction is to respond to it by resisting. That's because we're used to it being used to find when someone looks, in someone else's judgment, well, bad. But that's not necessarily the case. Why limit ourselves to defensive positions against "scary" technologies? Why not take some offensive ones? If computer vision can determine when we look bad, we can develop some computer vision technology that can figure out when we look good. And who looks better than... rock stars?

8 Coda: Keeping Things in Perspective

It's tempting to think of "sousveillance"[6] projects as empowering—but it can be a mistake. Although timidity in the face of surveillance is a risk, taking an active position presents the risk that we fool ourselves into thinking we've somehow changed the status quo. SVEN does nothing to disrupt authoritarian surveillance systems. But funny, even ridiculous examples can sometimes help break the ice and provide a way in to discussion of subjects that might otherwise seem dry, inaccessible—and scary. The author hopes that SVEN can help provoke rational discussion and understanding of the cultural and technical matters it addresses. Talk doesn't change anything either, but it can contribute toward a larger, mainstream shift in public perception—a shift in which the mainstream public doesn't see concerns about surveillance as limited to fringe activists, malcontents and other "scary" people. Similarly, tactical media projects with mainstream sensibilities could eventually make Big Brother resistance as popularly acceptable as the Big Brother TV show. Only through shifts in mainstream perception can we hope to see the disruption of scary status quos.

References

1. Bowles, W.: The Macintosh Computer – Archetypal Capitalist Machine (1987), http://www.williambowles.info/sa/maccrit.html
2. Hermetically Sealed Stuff is Magic. In: WikiWikiWeb, http://c2.com/cgi/wiki?HermeticallySealedStuffIsMagic

[6] *Sousveillance*, a term originally coined by Steve Mann, refers to community-generated surveillance activity: surveillance from underneath (sous) rather than from overhead (sur). [7]

3. Our Search: Google Technology. In: Google, http://www.google.com/technology/index.html
4. Johnston, D.C.: I.R.S. Limited Tax Refunds of Poor, Congress Is Told. In: New York Times (2006), http://www.nytimes.com/2006/01/10/business/10cnd-tax. html
5. Derieg, A.: Exploring Limited Spaces. In: Furtherfield Blog (2007), http://blog.furtherfield.org/?q=node/134
6. DeCoro, C.: Face Recognition using Eigenfaces (2004), http://www.cs.princeton.edu/~cdecoro/eigenfaces
7. Mann, S.: Sousveillance (2002), http://wearcam.org/sousveillance.htm