Coding, art, and kinetics. An interview with Joachim Sauter

*
Karla Brunet[1]
Joachim Sauter[2]

I met Joachim Sauter at UDK/Universität der Künste Berlin (University of the Arts) during my post-doctoral research year there. My main interest in this interview was to get to know the creative process of coding, especially programming codes developed by artists.

In the late 80s, Joachim Sauter founded ART+COM, an interdisciplinary group of artists, designers, scientists, and technologists focused on the computational art. Sauter has participated in many exhibitions in different museums such as 'Centre Pompidou' Paris,'Venice Biennial', 'Stejdilik Museum' Amsterdam, 'Museum for Contemporary Art' Sidney, 'Deichtorhallen Hamburg' , Kunsthalle Wien,'ICC' Tokyo, 'Getty Center' Los Angeles, MAXXI Rom.

Since the beginning of ART+COM, Sauter has worked in the intersection of art, design, and technology. He is also a full-time professor for "New Media Art and Design" at the 'University of the Arts' Berlin. Below, it is the transcription of an interview with Sauter at UDK in 2015. We talked about the way he deals with coding in his own artwork, his creative process and his thoughts on the relation of artists and programming languages.

*

[1] URL: http://karlabru.net/
[2] URL: http://www.joachimsauter.com

Figure 1. Photo by Gleb Leonov / Strelka Institute
Source: http://www.flickr.com/photos/strelka/27182305723/

KB. Your background is in art, is it in sculpture? You went to art school. What did you study in art? Sculpture?

JS. My background is first in design, visual communication and afterwards I studied film at the German Film Academy in Berlin. This was in the mid-80s. This was my background in terms of university education. But funny enough, when I finished the education, well I was educated at the time, for sure. Then, the first Apple Macintosh was on my table out of accident because a friend of mine was a computer dealer. it was really actually one of the first Apple Macintosh computer, this cubic thing, a 128k. This changed my life. Although I was very well educated, I more or less switched completely to make research about this new technology.

KB. Yes, that is my second question. When did you start with the technology and what was that? I was looking at your work, did your first projects were Internet-based or did you started directly with installation?

JS. It was installations. Actually, I started, as I said, in 84 but no one could predict this was a technology that was going to our lives as it changed afterwards. It was something that would change our disciplines. So I focused from that time more or less onto making research how this technology can be used in art and design. At very first, at the beginning, I used that technology as a tool. I did the same I did before but with this technology. From desktop publishing to animation, to imaging, and whatever... But then, let's say, mid-80s to end 80s it became clear that it was not only a tool but also a new medium. Because I was able to communicate and distribute information.

This was for me a turning point. Knowing that there is something that can radically change our way we communicate, how we are doing art. Also, I got my first email address in 86 so it was this more or less this insecure technology net which we had to research and we did already let's say net projects but more like the fax project in the times before. We were sending things to each other, changing it and sending it again. It was more the transportation. The idea the transportation rather than connecting things. With robots, it was the main quality of this medium. So it was this mingle between tool – getting the idea of a medium, starting to make the immaterial project but also physical installations. And this is what I am doing up to now, you know.

KB. Do you code? Can you program? In which language do you doc?

JS. So actually I started coding in the 80s with basic. Not visual basic, but basic even worst. Then afterwards I did a bit of everything so when Java came up and for sure I with this is the birth of Processing because Casey is a good friend. When I was at MIT I met him as a student there. When he and Ben started doing that and also with Arduino in Ivrea. So this was quite interesting to be always an observer and also a little bit a changer when things were coming out. And so having in one hand this very traditional education where, let's say, a Xerox – the copy machine – was the highest technology you could get hands off. And on the other hand seeing what happens with all this digital revolution. Yes, up to now it is more sketching but I'm not so heavily in coding because now I'm more into doing concepts having a lot of people working with me… I don't consider myself as a computational artist. I'm more an installation artist who is working with computation. I'm working in a group and having the specialist for that but with a deep understanding of what computation is.

KB. When did you start coding? What was the first kind of coding you did? You started with basic but why did you choose basic?

JS. It was more about an interest in how thinks were working. But 80 and up to the mid-90s have been a time where we call them artist/programmer couples. There were a lot of famous couples between an artist and a programmer, so 6 to 8. And I was also one of these couples together with Dirk Luesebrink as a programmer. Because what we had been able to do was with Basic, for instance, was more o less nothing. Drawing lines on a computer… So what we did was working together with a programmer but as a couple, you know, as an artist couple more or less. I said there were 5 to 6 couple of that time that more or less have been producing something for that new species. Then this [new programming languages], all the programming languages came up

which made easy for artists to program by themselves. And also the idea of programming became much more common. And nowadays, it's common. It's a common knowledge, a common technology. But as I said, in the 80s it was something only for specialists. There were some computational artists; I would say – Frieder Nake – who had been in the 60s and 70s trying out these machines. But the real computational design appeared in the end of 90s.

KB. One of my questions was if you see any difference from before to nowadays and how artist deal with code but you already answered me a bit about that. For example, in your projects how do you work with this? Do you collaborate with other artists, programmers, musicians, and engineers to create the artwork? How is the creative process of coding? Do you have an idea and then you get people to code for you, or you mess around with code and then come ideas. How is the creative process of this relationship of the artist and programmer?

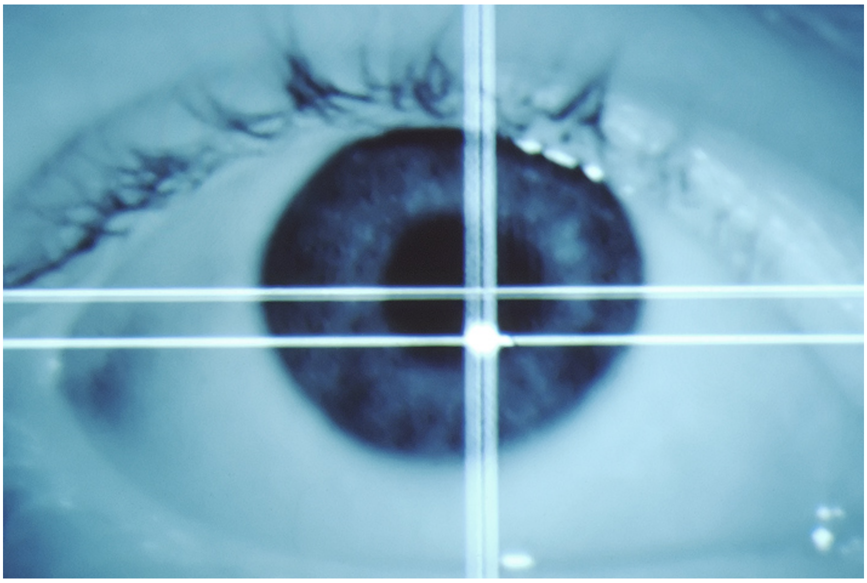

Figure 2. "Zerseher" - Fonte: http://www.joachimsauter.com/en/work/zerseher.html

JS. This is something… it is a mutual process. So, knowing about all the ideas about computation, we are creating projects out of it. It is all about knowing about the ideas of computation. And let's say the final aesthetical outcome is then done by someone who is really going to do the code. Let's say, for instance, very early projects which I did together with my programmer friend Dirk Luesebrink., it was the Zersher . It was something where you debut a picture painting with your eyes. And behind there was a very complex computation process, and we both discuss the process in terms that we didn't want to destroy the picture but we want to, let's say, distort the pictures in a way that every information stays where it is but representational changes the picture. So it is a computational idea which has been then executed by Dirk. But knowing about the ideas about the computational we could discuss together and afterwards work with it and process it. And this is up today. So, let's say, our process at art conferences there is a lot of sketching beforehand and sometimes it is given over to programmer although we do not really make a distinction between programmer and designer. Let's say, they are all media designer but probably with different backgrounds. These species is something which actually came out in the mid and end of 90 – the computational designer or the media designer, or the computational artists, media artists. Before it was the interdisciplinary thing. And let's say that the 80s and 90s had been the time of interdisciplinary team, now it is again the single artist.

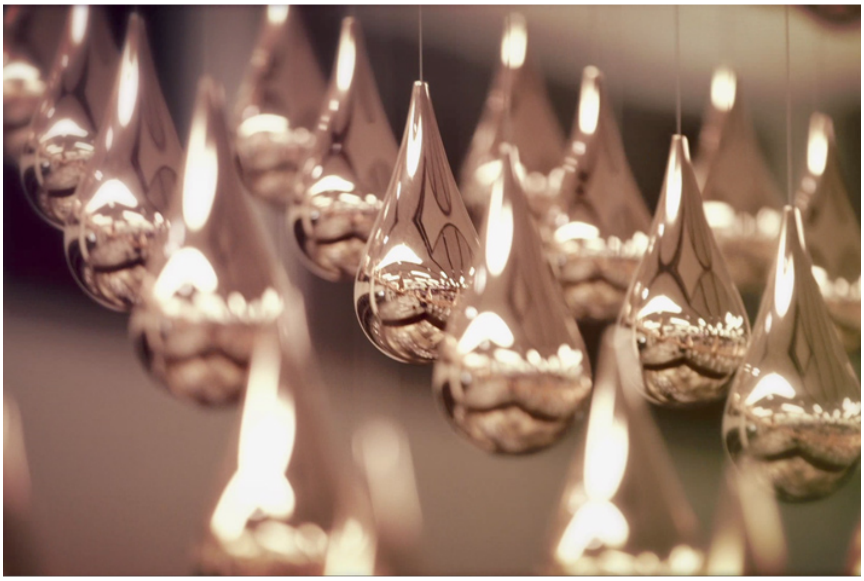

Figure 3. “Kinetic Rain” - Fonte: http://www.joachimsauter.com/en/work/rain.html

KB. In some of your artworks, for example, “Symphonie Cinetique” Or “Kinetic Rain” , at you don’t need only a programmer but also you need a lot of engineering background because of the structure and the way it works. How do you work with that? Do you have engineers working with you? How is the process? Is it something you can create the sketches but when you work with the materials, it changes a lot. You need the material, that’s why I asked if you had a background in sculpture. I believe you need a lot of knowledge on materials in order to develop these artworks.

JS. Actually, you know, of this long time of period we were doing this. Coming from graphic design and paper, going to film at that time with celluloid and chemistry and going then into installation you really learn quite quickly about all the ideas of the materials, of the physical materials. Then afterwards, you get started with the idea about physical computing... Then you get the idea of Mechatronics, and this is something you can learn in a way. Or you can inform yourself. And its always... the motivation comes always out of a project. So, the first kinect installation that we did it was for the BMW museum. There was this idea of creating dynamics of surface space. And there was no idea how one can create this because usually, you can’t have something floating in space. So we started to experiment with meshes or different kind of objects that we were hanging. And at the end, we came up with the idea of hanging ball on to it. Then the next step was how can we move them. Then it was clear we needed to look for a motor. Then we found out there were very precise motors out there which are used in factories for 30/40/50 years already. So that's all old technology but with a big difference, this is the main quality that we had the idea of computational design/computational art that we started to work with special controllers on the motors which we could then program them. Not only let’s say, beforehand people set a K, we had to put the motor in position A B C D, and we said ok, we do a computational, we do a code which then decides which position that comes at what moment of time. And this makes the quality; the difference to had been done before. So it is a very natural process. We have this kind of project, this idea, then it evolves and then you get inform yourself about things... And Then you find people who are specialist, and so... this is how things evolved. I don’t have any background in engineering.

KB. How you thing nowadays coding works, how art is coded? The way artists deal with the code has changed from the 90s. Because the 90s it was usually an engineer or a programmer together with an artist. And now you can see that is more a group thing or the artists are starting to code. What do you think is the cause of that?

JS. It is very natural because, let’s say, if you see the students’ generation now, they have been born in the early 90s. So when Internet came out. They are the first generation that more or less grew up with all the technologies and then – for them – it was very natural. Even now, at schools, you learn how to program. And nowadays there is also this easy to learn languages and easy access to the computer. Not like the first Apple Macintosh, it was 12 thousand dollars. This was impossible for a student. And nowadays everyone has the hardware, has access to easy programming languages and so this is how it appears. And it’s also a different kind of understanding if you grow up with this as a very natural thing. For us it was not natural, it was totally unnatural. It was something that was interesting and no one had seen this before. And nowadays it’s everyday thing.It is obvious.

KB. Since you teach here, nowadays how do you see art students’ coding compare to back then? Do you think art and design students are a lot more open to coding?

J***S.*** yes, we see even in the basic course here they learn to program. In the very first semester, they get this course and a lot of them they already are totally literate in them. So a lot of students you don’t have to teach the basics anyhow. In the meantime, it becomes a natural thing in the educational process in an art and design department. And so some of the students now wanna go into illustration or wanna work with pencil, and the other ones to other areas. If they have a big affinity for it [programming], to more media art and design classes and then they get very deep, they learn from each other. We have a course, a program it was in Processing but is more, let's say an addition. It’s where students get, on one hand, these mouth-to-mouth things if they have problems, but on the other, getting the idea of the field what can be considered as computational design. It's not only drawing something on the monitor, things like that.

So, they more or less learn from each other, this is what I figured out. And they also learn through being more or less thrown into it an idea of a project. This is the best motivation anyhow.

KB. Many of your artworks have a lot of interaction. They are not only installations but also interactive installation. Is that something it is important for you in your artwork? Or is it because technology enables you to do that?

JS. Interaction had been for me the most important thing in the 80s and 90s because there are two main qualities of that technology, the digital technologies. You can create a mutual dialogue that can create interaction. And the other thing is that you can network things, you can link things. And so going into that it was for me at that time the most important thing. These two things – networking and interaction – were the big thing. And then, let's say, computational came out in the late 90s and early 00s, you can observe the decrease of interaction in my work. And nowadays, it’s very few projects that I have interactively involved. It’s mainly computational kinetics or a lot of computational designs surfaces. Coded surfaces, this is the main focus now of the work. So its as I more interested in the development of the medium, my work follows more the interests what it is interested in the moment. As I said, in the beginning, it was more interaction and networking, now it is more computation. And the quality of the computational surface coded

KB. You mean the complexity of computation?

JS. Yes, in a way it is complex and... Complexity doesn't go against interaction but my interest actually lays more now in the non-interactive computational projects. So if you see the "Kinetic Sculpture" or the "Symphonie Cinetique", it is zero interaction. It's all about computation and mechatronics.

KB. Thank your very much