

# ANALYSING LIVE CODING WITH ETHNOGRAPHICAL APPROACH. A NEW PERSPECTIVE.

**Giovanni Mori**  
University of Florence  
giovanni.mori@unifi.it

## ABSTRACT

In this article, I will analyse live coding technique under the magnifying lens of Ethnography. Using this perspective, I will try to delve into three main aspects: the effect on the audience/performer interaction of the screen projection during performances; the relationship between “hacker’s ethic”, borrowing a Pekka Himanen’s definition, and live coders community; how some researchers are trying to establish contacts between formal and informal music milieu. In my view, an Ethnographical approach can help people interested in live coding to contextualise some implication of this technique’s internal dynamics. Additionally, this perspective can be useful to build a bridge between some academic and non-academic computer music contexts, using live coding as glue between them.

## 1. INTRODUCTION

A new tendency in Ethnomusicology is to include, among interesting experiences to document and analyse, not only the so-called traditional or folk music but also emergent contemporary music practices. Live coding is, without doubts, among these last ones. Many researchers think that is useful to deepen the knowledge of contemporary cultural phenomena to increase the understanding of present social processes and dynamics since their inception. The debate on this is still ongoing, though.

In the case of live coding, I think that an Ethnographic recognition made by the newest techniques can be very fruitful, because it appears to be a cross-practice music experience involving many different cultural contexts. Either the artist/audience interaction or the effect of screen displaying or, finally, the cultural milieu that lay underneath, may be interesting for understanding many social dynamics.

Then, in this article, I will explain the ongoing research I am carrying out for my PhD, trying to summarise the main concepts emerged during my inquiry both through “traditional” ethnological field documentation, through reading written documents and through the newest technique called “netnography” that I will explain better beneath.

## 2. LIVE CODING: A BRIEF DESCRIPTION

Live coding is an improvisatory artistic technique. It can be employed in many different performative contexts: dance, music, moving images and even weaving. I have concentrated my attention on the music side, which seems to be the most prominent. For playing with this technique, the performer have to use a dedicated software platform and a programming language to send, in real time, the music instructions to the machine. There are many live coding music programs and interfaces, but, with few exceptions, all of them are characterised by a central role of the text for managing the sound result. In fact, the computer software, for reproducing sounds, interprets the text typed in the program’s window. The performer express his virtuosity through high speed typing of complex instructions with the right grammar. Sometimes the musician can type program lines to see what happens, without having then a full control on the process. Therefore, there is a will of producing an indeterminate music results on a certain degree. This happens probably because we assist to an exploratory phase and the performer is aware that he/she cannot control the whole machine processes and in this way, it is possible to find some interesting but unpredictable music solutions.

There are also some other kinds of musical interfaces, as wearables or external devices, but all produce an alphanumeric result on the screen. Thus, text seems to be the main musical referent of this technique.

The musical material employed during events can be retrieved either from the performer's computer archive or it can be created from scratch, by using various techniques as, for example, additive or subtractive synthesis, filtering and so on. Computer is not utilised as a “traditional” musical instrument, with a cause and effect connection between the performer actions and the sound heard. Musicians put their musical thoughts in a written form during programming and then the computer software translates the code in sounds when every line is launched.

Live coding in music is practically a pure “instrumental” practice. With this term, I would point out the fact that in all the performances I have attended to and I have read about, there were not singers<sup>1</sup>. Nevertheless, live coded shows employ human voice samples, treating them in the same way of other musical material. In some cases, the human body carries out an active role during live coding performances. For example, there can be dancers, performers of different types of instruments or wearable electronic devices that track movements and other body features to transform them in data inputs.

Live coding concerts can take different forms. A constant, however, is that every performer shows, normally through a video projector, his or her computer screen to the public. This action brings the attendees inside the music making and in every single moment, they are aware of what is happening on the musician's computer. They do this for a matter of transparency: live coders do not wish to hide their musical actions because they feel themselves to be performers and they want that the public perceive clearly this. They do not want to be seen as someone simply standing behind a screen leaving the audience unaware of what he or she is doing. During many computer music performances, the artists may be suspected of checking emails or chatting with someone instead of playing, while the computer reproduces the music heard by the audience. Live coders think that with screen projection it is clear that they are playing and not doing something else. This act of transparency has probably something in common with conceptions coming from the world of popular music, where, during the last decades, authenticity, sincerity and transparency have become highly valued features (Auslander2008). The commitment in the art and the effort to obtain an “authentic” product and an “authentic” performance is typical, in particular, of underground popular music movements, less interested than major labels, at least in principles, in increasing their revenues exponentially. However, the need of authenticity from the listeners is becoming more and more urgent for pop stars too. See for example the Ashlee Simpson's lip-syncing scandal on Saturday Night Live, happened a few years ago, ended in some online petitions for convincing her to quit her career.

Quite underground as well, almost since the recent explosion of Android devices, is the so called “hacker's culture”, into which I believe that live coding can be included. I am using the term “hacker” to address the meaning given to this word by Pekka Himanen in his book *The Hacker Ethics and the Spirit of Information Age* (Himanen2003). Hacker's community is multifaceted but everyone shares some basic principles. One of the most important is that a good society has to be free and based on communities of pairs. A community works better and faster than a single alone in developing complex structures. The most famous example of this approach is Linux, the free Operative System *par excellence*. We should not interpret here the word “free” in economic terms. It is not a matter of cost but instead one of freedom. The code, in the hacker's view has to remain accessible; everyone should have the choice of accessing and modifying the code freely (Stallman2010). It seems that this belief is valid among live coders as well. Hence, there are also hackers who accept the employment of this working approach even in ICT companies for capitalistic purposes (Levy1984)<sup>2</sup>. As just hinted, live coders seem to adhere more to the hacker's movement branch, where it is important to share knowledge on equal basis, without requesting economic return for this, to give out intellectual products and let whoever wants to use and modify them freely. Another important aspect for hackers, as well as for live coders, is, finally, to build a community of passionate users devoted to the cause, whose feedbacks and suggestions contribute greatly to the developing of software. All the previous aspects are present in the live coding community. In fact, music programs are released under open source licence and the most famous program, Supercollider, is developed in a way similar to that of Linux. Moreover, there

---

<sup>1</sup> There are some rare cases of live coding coupled with human voice sung in real time, as for example Miko Rex or Silicon Bake.

<sup>2</sup> See also this interesting essay on hacker's culture based on Levy1984 but also on some other books on this subject (in Italian): <http://www.altrodiritto.unifi.it/ricerche/devianza/tavassi/cap1.htm>

are some online forums, as [toplap.org](http://toplap.org) or [livecoderesearchnetwork.com](http://livecoderesearchnetwork.com), where the community exchanges opinions and informs all the people interested in live coding on the group's activities.

On the live concert hand, it seems that there is not a live coding ideal context. In fact, live coding shows can take place almost everywhere, from nightclubs to concert halls. This happens because live coding is a performative technique and not a proper music practice. Then, musicians can produce many different music forms, timbres and dynamics characteristic of diverse kinds of musical manifestation by employing it.

There are probably some connections between the software's structure and the music produced with that. For example, Threnoscope program is more suitable to produce long and droning sounds and Tidal, Gibber, Sonic Pi, etc., instead, are more likely to be used in a collective event called *algorave* or, generally, in music contexts that request a bodily reception. However, it is possible to produce drones or long sound with Tidal or Gibber, but their constraints do not help in doing this. Threnoscope has a double sided interface: one circular that represent an ideal sonic round space where the sound is spatialised via multichannel sound system, and one based on text where the performer types the program lines that represent the instructions for sound reproduction. Considering that this program needs a complex sound system for expressing its complete creative power, it is more likely to be used in academic contexts, where music with similar needs is often staged. The Tidal or Gibber case is different, because their authors designed those simples, flexibles and able to manage sounds very quickly. Their interfaces has a unique window where performers write the program lines. The light interface and the intuitive syntax enable them to implement parameters with ease. In fact, these programs are utilised to reproduce more or less complex rhythmic patterns and melodies, and to modify both in an extremely rapid way. In the most part of performances I have attended to, the sound result was under many aspects similar to those of dance and techno music: stratified structures, repetitive beat patterns, high level of sound reproduction and so on. Thanks to these similarities, the programs are widely used in events called, as already mentioned, *algorave*. A rave is a music party based on concepts as that of freedom, equality, anticapitalism and, in general, extremely libertarian principles (Hakim2007). Such principles resonate also in the live coding community, even though with less extreme connotations and outcomes. In this article, unfortunately, there is no room to think about the relationship between live coding and rave culture extensively.

The most important software in live coding field, however, appears to be Supercollider, which has the most wide music possibilities. In fact, many different programs borrow many features from that as, for example, programming language, architecture and design. This program is divided in two parts: one visible to the musician (SCLang) and one invisible (SCserver). This last one is the machine that interprets in real time the instructions coming from the first one (the proper interface) and transforms them in sound. SCserver can be managed by using some other custom designed interfaces. In fact, many live coders use this piece of software through another program, which interact and exploit different SCserver features.

### 3. SOME REFLECTIONS ON LIVE CODING MUSIC PRACTICE

That was a brief, though not complete, description of some of the most important aspects of live coding performances. Probably now is time to reflect about implications of musical and performative actions taken inside this music community. However, before to begin the proper analysis, I would briefly introduce Ethnography, the inquiry method employed here.

Ethnography is characterised by three main rules: 1) direct observation and description of the social practices, 2) the research perspective is that of the researcher, that is partial, 3) this perspective is "full of theory" or, in other words, the researcher has a cultural baggage that influences his or her inquiry results.

Many techniques can be employed during an Ethnographic research. The most important in my research are the participative observation and the so-called netnography.

The first one entail a direct researcher participation in the cultural activities of a precise social group, for a sufficiently long period, going directly "on site". Ethnographer should establish a direct interaction with group's members to be able to understand and describe the purpose of people's actions in that particular context. In that way, knowledge is built through a process of identification by the researcher with the group's members (Corbetta1999). All this techniques are employed to obtain, firstly, an interpretation and description of a society *from the inside*, using the same words and the same concepts of "natives". Then, it

enable the “natives” to be conscious of their “tacit knowledge”: the embodied knowledge that people is not aware to own because they feel it as something natural. Finally, the researcher’s role should be emphasised because what he or she will tell depends on how he or she has conducted the field research.

The second technique to describe, Netnography, arise after spreading of Web 2.0. This term’s father is the sociologist and marketing researcher Robert Kozinets (Kozinets2009), who coined it to define a qualitative research<sup>3</sup> conducted on the “online” field. This technique enable the researcher to collect important behaviour information from the spontaneous chat of online users. The original aim of Netnography was to understand the consumers’ mood about commercial products, brands, prices and so on. With these insights, the marketing agency can construct personalised advertisement campaigns but sociologist too can extract important information on social interactions. Netnography has developed a scientific methodology to select useful data from the mass of online messages exchanged inside webtribes. A webtribe is simply a group of people, active on the social media, which speak about specific topics and creating a community around it. Netnographers employ non-intrusive and natural survey techniques, to limit the researcher’s interference with the normal field dynamics, letting the people to exchange opinions and evaluate products freely. Only in this way, researchers can have a faithful representation of people thoughts. These thoughts, in Netnography, undergone an interpretative analysis to extract from them both commercial and cultural insights. Commercial insights represent the people’s sentiment about a particular topic, how members employ a product and, finally, suggestions and complaints about something. Cultural insides include instead the study of “tribal culture” and “tribal ethics”: the first is how webtribe members represent themselves and how they see the world around; the second address the shared values among community members.

Although this technique has been developed for market research purpose, in my opinion it can be useful also to understand the live coding community members’ online activities because live coders as well is a webtribe grown around a “product” (the technique of live coding) that exchange opinions, evaluate, struggle to modify “the market” to match their needing.

Coming back to performance analysis, the most innovative aspect introduced by live coding, in my opinion, is the computer screen projection during concerts. This act has a clear scope: let the audience become aware that they are witnessing a live performance and not a fiction or an acousmatic concert. Some funny actions undertaken by performers, as asking through the screen questions to the audience, contribute crucially to construct this liveness effect. For example, Alex McLean has asked for a beer during his performance at the Lambda Sonic in Ghent, Belgium, typing his request on the computer screen and, consequently, on the projected image of his screen on the club’s wall. After a little while, the beer arrived brought by an audience member<sup>4</sup>. This weird request express the performer need not only for a beer during his show, but also, most importantly, for audience involvement on a different and unusual level. This appears to be a clear experiment of breaking the invisible wall between performer and listener during concerts. However, the main aim of screen projection is to show to those present the development of performer’s music thoughts in form of written algorithms and sound. Additionally, this entails that the musicians exposes themselves to the risks of every live performance. In most computer music concerts, the audience cannot be sure about the event's nature they are witnessing to, because calculator is not an acoustic music instrument, but acousmatic and potentially automatic. By “acousmatic”, I mean a context in which the sound source is impossible to recognise by direct visualisation (Schaeffer1966). In the computer music case, the audio output is the product of electric currents’ oscillation managed by software. Then the actual sound source is invisible and, especially in the computer case, the sound may be automatically driven.

A computer music concert may be compared, in some cases, to a sort of public records listening, with a marginal power of the musician on the sound output. In some other cases instead, the musician has a more important role, but what he or she is doing, is not evident and so it is barely distinguishable from the previous one on the visual aspect. What makes the difference between live coding and other computer music performances is that every live coder gives a proof of his or her crucial role as “event conductor”. The screen is the place where sounds take shape. Showing the ongoing music process is something comparable to play an acoustic instrument: the music making process becomes transparent and people in front of the

---

<sup>3</sup> A qualitative research use a definite group of research techniques for collecting precise information about a specific and limited case of study. A quantitative research, on the contrary, aims at collecting big amount of data on a wide research field and analyse them though the tool of statistics and mathematical models (Ronzon2008).

<sup>4</sup> See the performance’s video here: <https://www.youtube.com/watch?v=DmTuQcTTORw>

performer do not feel betrayed or cheated, but involved into the sound production. Performances can take place also via the internet, but this does not seem to influence the audience's liveness perception. It can be inferred that the public perceive the screen as the performer's real time representation, the performer's simulacrum. An example of this happened in Birmingham at the Network Music Festival. It was a telematic performance between three musicians. They interacted with each other on an online server and, through an Internet connection, they broadcasted the sound result in the concert hall. The audience, in attending at this performance, laughed at jokes, stayed silent during music "playing" and finally applauded as if they were in presence of the performer flesh and blood. This example confirms that in live coding concerts the attendees' centre of attention is the screen.

Therefore, it is evident that the performer has a crucial and active role in live coding performances. Nevertheless, as hinted above, it seems that musician's body tends to become less and less important, at least where the only sound production device is the computer. The listeners' attention is more focused on the screen, or better, on the screen's projection. This light beam appears to be the real music referent in this context and in fact, it has often the most prominent position in the performative space. People are attracted more from flashing projected images coming from computer than from the musicians themselves, because the screen conveys all the musical information that the performer cannot provide. This is also confirmed by the audience reaction when the concert is telematic. Apparently, the musician absence does not damage liveness perception. The interaction between performers on the server's chat window and with the audience confirm and reinforce this sensation.

All these statements seems to remain true even though the performative space changes and the music practice switches from a more rhythmic and informal context to a more static, reflexive and formal one. In fact, in performances like those made, for example, through Threnoscope or those taken by BEER ensemble, where audience members are not stimulated to move all their bodies following the music rhythm, the screen polarise the listener's attention probably on a higher degree. For example, during the performance held at Escola de Música do Conservatório Nacional in Lisbon by Thor Magnusson and Miguel Mira, my attention was attracted by actions represented on the screen projection and by those played by the cellist, who was very theatrical. In the same way, at algoraves my attention was polarised by the display's evolving images and not very much by the musician. The screen becomes a body extension, a medium of communication in itself and not only an interface. So, in the end, this action tends to eclipse the rest of musician's body. This is confirmed by, as I have already said, concerts played via the internet. Here musicians are somewhere else, but their screen is projected in front of us and becomes the only music referent.

Therefore, the screen projection is, in my opinion, the most important innovation introduced by live coding movement in computer music practice. However, there are at least another three important aspects to cite and analyse briefly.

The first one is that live coding has transformed the act of programming in an artistic performance. In fact, it seems to be one of the first experiences of transporting the software writing activity, which was until then a process of texting made *before* the real computer "action", in the real time domain. The verb "program" derives from Antique Greek *prò-gramma*, which means literally «to write before». Then, it is an action taken before the real event. With live coding is possible instead to introduce programming in many "on-the-fly" activities as music performances, but also many others. To become a widespread technique, it needs time to develop as science, to refine procedures and dedicated software.

Another important innovation is the introduction of hacker's ethic in music. I would like to introduce briefly what I mean with the word "hacker". First, I would like to state that people who define themselves hackers are not informatics criminals as widely implied by popular mass media. They are called crackers (crack + hacker). The term hacker addressed, at the beginning of computer era, a passionate person who likes to deepen the understanding of something by opening and exploring it directly, with his or her hands, and share this knowledge with other people to improve the community work. Only later, the hackers have started to switch their interest on computers. Even nowadays, the word "hack" alone address to an intelligent trick developed thanks to the deep knowledge of something.

During the movement's inception in the 1950, the first hackers coined the ruling principles of the group: passion, sense of community among members, desire to deepen personal knowledge and to share it with other group's members. (Levy1984). However, the community has changed with the technology developments. The computer has become an increasingly profitable business and this opened a fracture inside the movement. Some hackers tried to reconcile the aspect of gratuity, freedom and sharing

characteristic of the past, with that of business, causing some inevitable imbalances (privacy especially), and some others who remained faithful to the past principles. The two most famous hackers, Richard M. Stallman and Eric S. Raymond, represent two main tendencies in the community. The first is the most faithful to the original hacker's principles and he is a champion of activism for freedom of knowledge, users' privacy protection and community building<sup>5</sup>. The second is more right-wing libertarian and, even though he recognises hacker's ethic principles worth to be followed, he is less convinced about the importance of activism and more flexible about the basic principles' interpretation. He is also a very controversial person, condemned by many hackers for his ideas about the right to own weapons without limitations, about his ideas that this community should not be afraid of dealing with big ICT corporations as Microsoft and some other extremely right-wing ideas<sup>6</sup>. However, this article is not the appropriate space to talk about hacker community internal dynamics extensively. I would only to paint a little portrait of it and to place live coding community inside that. Apparently, it seems that live coders stand more to the Stallman's side than to the Raymond's one. Nevertheless, I have not had the occasion to delve into this aspect yet. It will be one of the subjects of my future research.

To come back now to live coding, I wrote in another article that there were previous examples of similar working method employed for musical purposes, but this happened before the hacker's movement birth, so probably it is not correct to label it as "hacker". However, live coding is the first music milieu in which recent hackers' lesson seems to be recognised and taken as example among its members. The most famous example of this lesson is the making of the Linux operative system by Linus Torvalds. Before live coding inception, there were only a few possibilities to modify and adapt the music instrument to every single personality. One of these methods was to build personalised instrument, either by artisans or by DIY techniques. However, every piece of them has its own constraints and affordances, especially in terms of pitch range, timbre modulation and number of simultaneous voices. With programming, all these obstacles are pushed away, or better, anyone can construct his or her preferred instrument affordances. Before live coding's inception, music was made quite exclusively by commercial music software. This kind of software is usually protected against improper usage and, by consequence, against the violation of original code. Then, the only solution to have customizable music software, apart from make illegal actions, is to employ free music software that is, by definition, freely "hackable" by anyone who wants to do it. Probably, some software engineers can have tried to develop his or her customised and customizable music software before, but they had to start writing it from scratch, because they did not had any pre-existent music program to modify. Live coding appears to be the first music movement that has employed hacker's approach to music software: reuse and rewrite existent programs and adapt them to personal needs. Hence, it has greatly contributed to this new music thread by developing many open source music programs and by adopting a collaborative and informal working method among the community members.

The last aspect to analyse about live coding is the relationship between popular and academic music. Considering that this technique is often employed among dance music practices, especially in Britain but also in North America, Germany and Spain, it has raised a question of how dance music is perceived in academic contexts. Alex McLean and Adam Parkinson (Parkinson and McLean 2014) have recently posed this problem in a conference speech, explaining that dance and academic music is frequently seen as opposed, even though they share, in McLean and Parkinson's opinion, some features and approaches. I do not want here to synthesise the whole article content, but only to emphasise the consequences of a particular statement: «From a research perspective, failure of an algorave can be illuminating. For example, if we find ourselves standing in a room looking at each other, issues of gender disparity which gravely undermines computer music culture becomes difficult to ignore». This couple of phrases addresses a very problematic situation that affects many university departments but not so much the audience in the dancehall. In the article, they demonstrate that this is not a live coding derived issue, but an academic one. In fact, when they have promoted algoraves through standard dance music promotion channels (radios, magazines etc.), says the previous authors, they attracted many non-academics that enjoyed live coded music, without gender distinctions. Nonetheless, gender imbalance remain an issue on the stage, where, both in live coded music and in dance music in general, the event's line-up is very often a list of masculine names with rare interruptions. This is evidently a general problem of many Western societies, not attributable to live coding. However, the McLean and Parkinson paper, in posing this issue, signal that the

---

<sup>5</sup> See Stallman's personal website: <https://stallman.org/>

<sup>6</sup> See Raymond's post on Linux Today:

[http://www.linuxtoday.com/developer/1999062802310NWSM#talkback\\_area](http://www.linuxtoday.com/developer/1999062802310NWSM#talkback_area).

community perceive this imbalance and is reflecting to find possible solutions. To conclude, we can state that live coding, as cross-practice technique, can be used both in academic and dance music and so it can put this two worlds in contact and pose problems localisable on the edge between the two. It can become a bridge for connecting these two apparently distant worlds.

#### **4. CONCLUSION**

In this article, I have illustrated the most important features that live coding technique has brought in the computer music performative context. After a presentation of these innovations, where I spoke of improvisation, the screen projection, algorave context and so on, I tried to go a little bit under the surface, interpreting the effect of these new introductions in computer music performances. All this subjects have been discussed after an Ethnographical research, which is the paper's main information source.

I have especially emphasised the aspect of the screen projection, saying that this beam of bright code becomes a sort of musician's body substitute, because it seems that the audience is more attracted by this than by the musicians themselves during performances. The code on the screen, then, becomes the real sound source in the public's view and it is interpreted as the alphanumerical representation of performer's musical thoughts. It may be interesting to interpret this situation with a baudrillardian simulacrum's theory perspective, but here it is only possible to hint at this choice. It may be an interesting subject for a future article.

Among important things said, I would like to mention the relationship between live coding movement and "hacker's culture", which appears to be more than a simple inspiring source. In fact, many principles adopted by live coders, as the development of free programs, the building of an equal community and a collaborative approach, are shared between the two. Probably it can be appropriate to live coders as music hackers.

Finally, I have illustrated how live coding community reflects on the relationship between academia and informal musical practices as that defined "dance music". I have observed, referring to the McLean and Parkinson's article, that live coding can connect these two worlds, considering its double-faced nature of improvisatory technique developed in an academic milieu, but used both in formal and informal music contexts.

Therefore, to conclude, I think that this brief article, thanks to the Ethnographical approach used to collect information, has demonstrated the multifaceted nature of live coding technique and that it has begun to shed light on some important aspects of audience/performer interaction that may have an influence on the future of computer music performance's developments. Ethnographical inquiry method has resulted to be a crucial tool to collect information about the community members, its internal dynamics and to develop a comparative view between live coding and other artistic and cultural contexts.

## REFERENCES

### Books and articles

- Auslander, Philip 2008. *Liveness: Performance in a Mediatized Culture*, New York, Routledge.
- Collins, Nick, McLean, Alex 2014. "Algorave: live performance of algorithmic electronic dance music". *Proceedings of the NIME conference*.
- Collins Nick et al. 2003. "Live Coding Techniques for Laptop Performance", *Organised Sound* 8(3): 321-330.
- Corbetta, Piergiorgio 1999. *Metodologia e tecniche della ricerca sociale*, Bologna, Il Mulino.
- Cortellazzo, Manlio, Marcato, Carla 2005. *Dizionario etimologico dei dialetti italiani*, Torino, Utet.
- Hakim, Bez 2007. *TAZ. Zone Temporariamente Autonome*, Milano, Shake.
- Himanen, Pekka 2003. *L'etica hacker e lo spirito dell'eta dell'informazione*, Milano, Feltrinelli.
- Kozinet, Robert 2009. *Netnography: Doing Ethnographic Research Online*, Los Angeles, Sage.
- Levy, Steven 1984. *Hackers: Heroes of the Computer Revolution*, New York, Delta.
- Magnusson, Thor 2014. "Improvising with the Threnoscope: Integrating Code, Hardware, GUI, Network, and Graphic Scores", *Proceedings of the NIME conference*.
- McLean, Alex, Sicchio, Kate 2014. "Sound choreography <> body code", *Proceedings of the second conference on Computation, Communication, Aesthetics and X*, pp. 355-362.
- McLean, Alex, Parkinson, Adam 2014, "Interfacing with the Night: intersecting club music and academia", *Proceedings of Inter-Face: International Conference on Live Interfaces*, 2014.
- Raymond, Eric S. 1996, *The New Hacker's Dictionary*, Cambridge MA, The MIT Press.
- Ronzon, Francesco 2008, *Sul campo: breve guida pratica alla ricerca etnografica*, Roma, Meltemi.
- Shaeffer, Pierre 1966. *Traité des Objets Musicaux*, Seuil.
- Stallman, Richard 2010. *Free Software, Free Society: Selected Essays of Richard M. Stallman*, Boston, GNU Press.

### Websites

- XIX International Seminar of Ethnomusicology, Jan. 30th - Feb. 1st 2014: <http://www.cini.it/events/xix-seminario-internazionale-di-etnomusicologia-living-music-case-studies-and-new-research-prospects>
- Alex McLean's project on live coding and weaving: <http://kairotic.org/making-a-warp-weighted-loom/>
- Algorave official website: <http://algorave.com/>
- Ashlee Simpson lip-syncing concert: <http://www.veoh.com/watch/v15490419bqh6yfdc?h1=Ashlee+Simpson+lip+sync+screw+up+on+SNL>
- Discussion on one Kraftwerk concert: <http://createdigitalmusic.com/2015/02/kraftwerk-live-rig-exposed-really-checking-email/>
- Discussion on some aspects of hacker's culture: <http://www.altrodiritto.unifi.it/ricerche/devianza/tavassi/cap1.htm>
- Eric Raymond's post on Linux Today: <http://www.linuxtoday.com/developer/1999062802310NWSM>
- Gibber official website: <http://gibber.mat.ucsb.edu/>
- Ixilang official website: <http://www.ixi-audio.net/ixilang/>
- Live Coding Research Network's website: <http://www.livecodenetwork.org>
- Marije Baalman's project: <https://www.marijebaalman.eu/?p=16>
- Richard Stallman's official website: <https://stallman.org/>
- Sonic-pi official website: <http://sonic-pi.net/>
- Supercollider official website: <http://supercollider.sourceforge.net/>
- Petition against Ashlee Simpson: [http://www.boston.com/ae/celebrity/ashlee\\_petition/](http://www.boston.com/ae/celebrity/ashlee_petition/)
- Tidal official website: <http://yaxu.org/tidal/>
- Toplap blog website: <http://toplap.org>