Kinematograph

Dr. H. Rodriguez, BA, MA, PhD.
School of Creative Media, City University of Hong Kong, Hong Kong
sweb.cityu.edu.hk/theblackbox
e-mail: smhect@cityu.edu.hk

Abstract

Kinematograph is an artistic research project that raises fundamental questions about the nature and direction of computational and algorithmic art. Instead of approaching information technologies as mere tools, this project regards them as conceptual structures that can be critically examined as such. The essential methodology is to begin with a well-established algorithm, in this case the Huffman compression scheme, which is not customarily considered as an artistic medium. Instead of treating this algorithm as an efficient tool for lossless data compression, however the Kinematograph deploys it as the basis for an aesthetics of dispersal and defamiliarization. This aesthetic venture refuses to endorse the ideology of power and control endorsed by many contemporary theories of interactivity. The search for power is to be replaced with a problematical and experimental questioning of technology in the most radical possible way.

1. Problematical questioning

Kinematograph is an ongoing artistic and theoretical project in computational art. In its critical engagement with information technologies, it keeps faith with philosopher Gilles Deleuze’s suggestion that the future of cinema depends on “its internal struggle with informatics” [1]. This statement can be extended, perhaps, to all of contemporary art.

In Kinematograph, the struggle takes the form of a question: “How does one do informational art?” This question does not call for a definite response. Its aim is not to arrive at some absolute answer that will dispose of the question once and for all. Rather, the aim is to render the question itself problematical [2]. To tackle the question is to learn how to pose the question anew. This process can be described as “experimental”, but the term can be misleading. An experiment is sometimes understood as a technical setup oriented towards a predefined range of possible answers. But the questioning involved in Kinematograph does not aim to select the right answer out of a pre-given set of possibilities. Just what might count as a possible answer is itself part of the problem. Heidegger famously noted that thinking “is a way”, but not in the sense of a well-worn route between two points. It is not a path that pre-exists the wayfarer’s journey. Thought exists only in the very act of raising the question, only so far as we remain “underway” [2]. To advance along the
way is to engage in an ongoing practice of problematical questioning. *Kinematograph* is experimental in this radical sense.

The media artist begins the journey by acquiring a certain facility in the use of informational technologies. Know-how is essential to the computational artist. But here, at the very outset, the question itself already becomes problematical. What does it mean, to learn information technologies? Many artists choose to learn a higher-level programming language. Some also attempt assembly language or electronics. Others study the mathematical theory of communication, as well as cybernetics, information systems, or computer graphics. The fundamental definition of the domain is not given in advance. There are multiple entry points. We do not know what information art consists of before embarking on the journey. And yet we must make decisions about what to do, which apparently require at least a preliminary understanding of the subject matter. To begin is to make an initial choice.

It would be a mistake, however, to describe the subsequent journey as a progressive elaboration of some initial definition or plan. The way is neither continuous nor linear. It proceeds by delays, detours, regressions, doubts, and radical revisions. We may even come to discover that we never knew what information was, and so must reconsider our fundamental assumptions from scratch. The best visual metaphor for this process is the well-known image, proposed by Paul Klee in his educational notebooks, of "taking a line for a walk". The journey resembles a line that meanders without a fixed direction, lingering and folding back upon itself.

2. The Outside

I started this project by selecting a well-established technique that is not customarily viewed as an artistic method. The technique chosen was the Huffman compression algorithm. The journey begins, quite simply, by learning how to implement this algorithm (in this case, using the Java programming language). The reason to select this particular technique is that Huffman encoding was first published in the early 1950s, when the foundations of the mathematical theory of communication had only recently been laid out. The design of the Huffman algorithm directly exhibits many of the assumptions at the heart of this then emerging field of research. More crucially, artists do not normally consider Huffman compression as an arena of creative exploration. Thus the starting point of the journey already involves a confrontation with an "outside".

This alien quality is an important feature of the way of working being proposed here. The process starts out by stepping outside of art, in the sense that the object of exploration belongs to scientific or engineering fields that have traditionally been excluded from artistic education and discourse. This is not an arbitrary decision, but rather a core aspect of the subject at hand. The artist who chooses to learn computer programming thereby enters a domain customarily regarded as "technical" and so foreign to art. This estrangement is an important stage in the way of the information artist, and the *Kinematograph* project embraces it head on.
This decision runs counter to dominant tendencies in media art education, where students are encouraged to learn digital tools by developing graphical or sound applications that are obviously "artistic". An animator might, for instance, start out with a definite visual outcome, such as drawing a perfect sphere or a realistic image of water, and then develop computational applications to realize this outcome. The relevance of technology to artistic applications is accepted from the outset. But I would suggest that any effort to smooth over the tension between art and technology tends to fall back on standard ways of using digital computers as artistic tools. Alternative possibilities are seldom explored.

Instead, the Kinematograph project commenced without any definite outcome in mind. It was not clear in the beginning whether its final output would be a graphical image, a sonic installation, an interactive application, etc. I regarded that act of learning the Huffman algorithm as an end in itself. This way of commencing the journey already implies huge risks, since it was not at all clear whether the process would eventually result in anything that could even remotely be classified as art. The Kinematograph project thus embraces thinking as an open adventure. It treats the field of "art and technology" as the site of an ongoing uncertainty.

3. The black box

This way of working is unusual, because technology is often regarded as a tool, and a tool is defined as a means to realize some predefined end. Considered as a tool in this sense, technology takes the form of a black box. Vilem Flusser's study of photography suggests that the concept of the black box constitutes a core aspect of media art [3]. Every digital technology essentially appears as a black box. A black box contains input and output terminals. To know how to use the black box is to know how to select the inputs that will reliably produce certain expected outputs. The user need not, and often cannot, know the internal mechanism that connects inputs and outputs. This opaqueness defines the black box.

The concept of the black box implies a dialectic of mastery and exclusion. The user learns to "master" the tool, and yet this mastery does not lead to understanding. The tool remains opaque. This dialectic manifests itself, for instance, in interface design.

To design an interface is to establish a context of relevance for the "user". According to this ideology, the task of good design is to shape the user's mental model of the device by supplying clear analogies or metaphors. Such metaphors should be obvious to everyone; they should also express clearly and unambiguously the core functionality of the device. The aim is to communicate the intended function of a device in its external design, which should be based on cultural analogies and natural models that are immediately understood. But the face presented by the machine to the user typically has no visible connection to the formal implementation. The interface, according to many designers, should have no relationship to the underlying structure of the device. The user's ignorance of the internal structure of the device is
thus taken as a premise of the design methodology.¹

The problem lies in the assumption that the device must present a friendly “face”, which must accommodate the user’s prior knowledge and experience. Interface design, as normally conceived, is regressive. Instead of reinforcing natural analogies and immediately comprehensible metaphors, the Kinematograph proceeds by inventing new ways of thinking and acting. I propose that we view informational arts as an experimental process of interrogating technologies. In the process, artists might discover new interfaces that illuminate and problematize the relationship of the user to the black box.

4. Technology as a way of thinking

To problematize technology means, first of all, to treat it as a style of thinking. What does this mean? It is common to view information theory as an “engineering” paradigm. There is obviously a great deal of truth in this assumption. The concept of information was meant to facilitate practical tasks pertaining to data storage and transmission. The Huffman algorithm, in particular, was designed to supply an optimal method of data compression. Its underlying motivation was essentially pragmatic. The Kinematograph project does not, however, consider information theory exclusively as a tool designed to solve pre-existing engineering problems.

Information theory is first and foremost a conceptual construction. The theory supplied an innovative vocabulary that has transformed our thinking about the tasks to which it is applied. Terms like “source”, “addressee”, “message”, “communication”, and “channel” acquire new technical meanings in the context of the theory. These meanings have essentially modified the basic terms of the engineering problems that the theory was designed to tackle. Instead of considering technology instrumentally, as a means to some end, the Kinematograph project aims to characterize technology as a frame of mind, a style of thinking. The basic move, in other words, is to isolate the theory from its intended application domain (its engineering context) by taking it as a conceptual edifice that can be explored as such.

This conceptual construction is hidden from the user of the black box. To use the black box is to ignore the ways of thinking that made possible the fabrication of the black box. My approach therefore outlines one way of subverting the black box concept. By treating technology as a style of thinking, moreover, I propose a way of relating to the digital that avoids two bad alternatives: uncompromising refusal and blind acceptance. The former fails to explore the novel ways of thinking and acting that can potentially arise from human-computer interaction. The latter normally celebrates the role of the user, capable of deploying digital tools without understanding their internal mechanisms. There is another choice, which proceeds

¹ “Software design is the act of determining the user’s experience with a piece of software. It has nothing to do with how the code works inside, or how big or small the code is.” Terry Winograd, Bringing Design to Software, Addison Wesley, 1996, chapter 2. http://hci.stanford.edu/bds/2_liddle.html
by critically examining the fundamental concepts of information technology. This is the choice adopted in the present project.

The guiding question is: how does information theory, and the Huffman algorithm in particular, define the conditions of its field of problems?

5. Formalism

Information technology is (broadly speaking) formalist. It concerns itself exclusively with the topic of symbol manipulation. Questions of truth and meaning (“semantics”) are outside its domain. Information theory defines its object in formalistic terms. The entire theory depends on this fundamental exclusion of content. Its focus pertains to the tasks involved in sending symbols over some channel, such as for instance telegraph or telephone lines, regardless of what those symbols happen to mean.

The *Kinematograph* project embraces the formalist problematic and disregards all questions of semantic content, thus situating itself firmly within the conceptual field of information theory. This formalist orientation manifests itself in the underlying concepts of the theory, such as “source” and “message”. An information “source” is an agency that makes a selection out of an ensemble of possibilities, such as for instance a team leader who picks a team member to perform a task or a novelist who selects a letter to write down on paper. All of these actions involve a choice from some set of possibilities. The source need not be an intentional agent; it can be a natural process or an artificial machine. From an informational standpoint, the meanings attached to the choices made by this source are irrelevant. What matters is that somehow a selection is made, and this selection reduces the number of possibilities to one.

Information theory always regards every “message” (choice) in relation to an ensemble of possible messages. This raises the important question of the criterion for the identity of messages. In other words, what counts as the “same” message? Suppose that one source chooses one out of four possible letters (for instance, A, B, C, and D), whereas another source chooses one letter out of the entire Roman alphabet. Even if both sources happen to choose the “same” symbol, for instance “A”, they do not thereby produce the same message. The reason is that the second source has more possibilities to choose from. Two messages are from this standpoint different, even if they are perceptually indiscernible, because they have been selected from a different set. The statement that one symbol has been selected is only informative relative to a set of possible choices.

6. The Dictionary and the Trace

The idea of an ensemble of possible choices can be formalized in the following way. Suppose that a team leader needs to select a team member to go on a difficult mission. She can make her selection using the following method: first divide her
entire team into two disjoint groups, then throw a coin to select one group, and iterate
the procedure until the chosen group has only one member. This technique
comprises a sequence of binary (yes/no) decisions. If the team has only four
members, the leader need only make two binary decisions to reduce the possibilities
to one [4]. Every decision can then be assigned a binary digit (0 or 1). The resulting
binary code can be understood as the static representation of a temporal process:
the progressive reduction of the set of possible choices by means of a sequence of
yes/no decisions.

For instance, suppose that the members of a team are called Ahmed, Maria, Jamie,
and Kenji. The tree in Figure 1 represents every possible sequence of binary
decisions.

![Binary Tree Diagram]

[Figure 1]

Given a finite ensemble of possible messages, it is always possible to represent the
entire network of binary choices available to a source by means of such a binary tree.
Every possible message is a terminal node. Every non-terminal node represents a
binary decision. I propose to use the term “dictionary” to describe such a binary tree,
which characterizes the entire network of binary choices for some ensemble.

The dictionary illustrated here can be used to identify every team member uniquely.
To determine the binary code for a particular person, simply proceed downwards
from the root towards the person in question, adding a 0 to the code when choosing
a left node or a 1 when choosing a right node. Thus the codes for Ahmed, Maria,
Jamie, and Kenji are 00, 01, 10, and 11 respectively. Another well-known dictionary
is the ASCII system of character encoding. Every symbol is encoded as a sequence of
seven bits, which means that seven binary choices are needed to reduce all
possibilities to one. The binary code for the symbol “A”, for instance, is 1000001.

From this viewpoint, then, a binary code is the static expression of a temporal
process. I propose to use the expression “binary trace” to describe the representation
of a dynamical process as a binary choice sequence. This brings me to an important
methodological rule: always understand every static digital representation as the
residue of a (real or hypothetical) dynamic process. I call this precept the "temporalization rule", because it suggests that the basic information theoretic concepts are essentially time-based. A binary sequence is of course a static object, but I propose to view it as the frozen picture of a dynamical process, the progressive reduction of an ensemble of possibilities by a sequence of binary choices. The temporalization rule suggests a tentative connection between information theory and the "time-image" that Deleuze identified as essential to cinematic and perhaps also media art [1].

7. Ordering

The method of successive binary decisions is of course only a fiction. Actual sources seldom actually proceed by successively partitioning the possible messages into two disjoint groups. This fiction, however, highlights an important conceptual point. To transmit the information that (for instance) one out of four elements has been chosen, the source only needs to write down a sequence of 2 binary digits, since every element is uniquely identified by one such sequence. For instance, if the team leader wants to notify another team that Maria has been chosen for some mission, the leader can transmit the code 01 via some communication line. The team on the receiving end can then use the appropriate dictionary to decode the information received.

This example shows the extent to which the practical problems of data transmission constitute the horizon of information theory. The theory was intended to facilitate the task of relaying messages encoded in binary symbols. The engineering aspects of information theory are evident, above all, in its concern with issues of fidelity and economy: How can messages be transmitted without losing information (fidelity) and with the shortest possible average code length (efficiency)?

In 1952, David Huffman invented an algorithm for the construction of a dictionary that generates a maximally efficient encoding [5]. The dictionary is organized by associating a frequency or probability with each of its elements. It is straightforward to discover the frequency of every message by recording the actual choices made by some concrete source over an arbitrarily long span of time. Suppose that, over the course of two months, the team leader selects Kenji 20 times and Jamie 10 times. Each of the remaining two members, Ahmed and Maria, is only chosen once. It is convenient to represent the behaviour of such a biased source as a binary tree, where the more frequent messages are placed closer to the root. In other words, the proximity of a terminal node to the root depends on the probability that the source will select the team member represented by this node. The hierarchical organization of the tree thus manifests the "preferences" or "biases" of the source. Every node now has a number associated with it, the frequency with which the team member represented by that node has been selected. The dictionary of Figure 2 captures this idea.
The outcome of the Huffman algorithm is that every message now has a variable (rather than fixed) length encoding. As illustrated in Figure 2, the codes for Kenji, Jamie, Ahmed, and Maria are now 0, 10, 110, and 111 respectively. The number of binary digits that encodes a message depends on the probability that the source will select that message: the greater the probability of a message being chosen, the shorter its binary encoding.

Suppose that our team leader has to transmit her selections over some communication channel. Her choices are: Kenji, Kenji, Kenji, Jamie, Jamie, Kenji, Ahmed, Maria. The code for this sequence of messages, based on the scheme shown in Figure 1, should be 1111111010110001. The new scheme, in contrast, yields the shorter binary string 00010100110111. The reason for this compression is that the codes of the more frequent messages are shorter than those of the more infrequent messages. If every message occurs with equal probability, the Huffman method does not offer any advantage over the fixed length encoding illustrated in Figure 1. But if the source displays some bias, the codes can be considerably reduced. A binary code is “compressible” or “redundant” if an algorithm exists that can reduce its length (measured in bits).

The efficacy of this algorithm is well known. I propose, however, to bracket the engineering aspects of Huffman’s technique, and to concentrate instead on the style of thinking that underpins it. Although it is not necessary here to give a detailed description of the algorithm, it is worth noting that the procedure builds the tree by first arranging every possible message into a queue. The Huffman algorithm essentially depends on the assumption that all possible messages can be queued in

---

2 Moreover, the code for every possible message is not the prefix (the first part) of any other code. This property ensures that messages can be decoded at the receiving end without any ambiguity. For example, 0 is the code for Kenji, and this is not the beginning of the code of any other team member.
order of frequency (or probability). This presupposition, which underpins the construction of the binary tree, highlights an important feature of every information system: It consists of a dictionary whose elements can be arranged into some definite order. In this case, the ordering is based on the frequency or probability associated with every possible message. The essential function of frequencies within an information system is to render messages quantitatively comparable, so that they can be arranged in order. Describing the origin of modern technology, Heidegger aptly notes: “nature reports itself in some way or other that is identifiable through calculation and that it remains orderable as a system of information.” [6]

Certain additional features of Huffman encoding play a key role in the current version of the Kinematograph project. The Huffman dictionary can be altered dynamically. A sequence of new choices made by a source whose behaviour changes over time will alter the frequencies associated with every message, and so modify the configuration of the Huffman tree. Suppose that our team leader begins to choose Ahmed far more frequently than before. This element will have to be brought closer to the root of the tree, and so the entire structure will have to be recalculated. Whereas a scheme such as ASCII assigns an unchanging code to every character, a Huffman dictionary is not necessarily fixed once and for all. For this reason, whoever receives the coded message must have the Huffman dictionary in order to restore the original message. Without access to the Huffman tree that was used to encode the message in the first place, it cannot be appropriately decoded. This point is important in this context, because the current version of the Kinematograph deliberately hides the underlying dictionary from the users of the work.

8. Installation description

The current version of the Kinematograph is an interactive installation that addresses the various aspects of information technologies mentioned here. The core application was originally developed in Java and subsequently extended in Processing. The basic setup consists of a five-screen projection system controlled by five networked computers. The interface to the entire system is a single keyboard. Exhibition visitors interact with the system by typing. The computer stores frequency data (i.e., the number of times that visitors choose to press a given key) in the form of a Huffman dictionary. The dictionary is continually updated in response to every new key pressed by the visitor. The five screens visualize the changing dictionary. Thus the visitor is both the source and the addressee of the messages. But, instead of displaying the messages that were typed, the five displays foreground the operation of the underlying algorithm. The installation thus ignores the main practical motivation for Huffman encoding: the faithful reproduction of information. Instead, it develops a cognitive laboratory.

The first screen visualizes the dynamical construction of the Huffman tree in response to the characters typed by the visitor. The visualization is based primarily on Paul Klee’s famous description, in his teaching notebooks, of taking a line for a walk. The display consists of two lines, one red and one black, each moving in a stepwise manner (Figure 3).
The black line displays every binary digit in the standard ASCII code for the key typed by the visitor; if the bit is a 0, the line moves horizontally; otherwise it moves vertically. When reaching one of the edges of the screen, the line continues moving in the opposite direction. When the last bit of the code for this particular key is reached, the system computes the following mapping:

\[
\begin{align*}
\text{int } x &= \text{map( int, } 0, 122, 0, \text{ width }) ; \\
\text{int y} &= \text{map( int, } 0, 122, 0, \text{ height }) ;
\end{align*}
\]

The line moves to the point with coordinates \((x,y)\) on the screen, and continues its motion as soon as the user types a new character. The aesthetic effect is of a line that meanders all over the two-dimensional screen. Its rapid motion visually expresses the underlying idea that a binary code is the trace of a dynamical, essentially temporal process.

The red line uses a similar rule to visualize the Huffman encoding of the character typed by the visitor. Its motion will be somewhat slower and less complex, particularly for characters which are very frequently typed, since the average length of the Huffman code tends to be shorter than that of the ASCII code. The graphical movement of the two lines thus visualizes the essential properties of the underlying algorithm. Visitors often experiment with the system by, for example, typing the same key over and over to observe the effects of their inputs on the visual representation. Any subsequent change in the visitor’s behaviour, for instance, a modification in the frequencies with which certain characters are typed, causes a sort of “perturbation” of the system. This alteration forces the software to reconfigure its underlying dictionary, which in turn affects the movement of the line. The kinetic “style” of the line thus responds to the changing behaviour of the visitor.
The second screen displays the ASCII code of every key pressed by the visitor, but the display relies on a special typography created specially for this project. Each symbol consists of up to seven vertical and horizontal strokes. Every stroke visualizes a single bit in the ASCII representation of the key pressed by the visitor. Thus, for instance, the binary encoding of the small "a" is 1100001. In the graphical representation of this letter, only three of the seven possible strokes are shown. The binary encoding of the capital "A", in contrast, is 1000001, and its visualization in my alphabet accordingly has only two strokes (Figure 4):

![Image of the letter "A" generated from ASCII code.]

[Figure 4]

The characters are displayed in the fashion of classical Chinese writing, from top to bottom and from right to left. This graphical arrangement is arbitrary, but it expresses how the structure of Chinese writing, where every symbol can be decomposed into basic elements, loosely inspired the design of this alphabet (Figure 5).

![Image of the Chinese characters alphabet.]

[Figure 5]
The resulting alphabet is of course opaque to first-time users. Its strangeness dramatizes the condition of the user of a technological black box, for whom the internal operation of the device is essentially obscure. It is nonetheless theoretically possible for the visitor to "learn" this encoding, since every key is consistently displayed as a clearly recognizable symbol. By typing the same key over and over and observing the visual outcome, for instance, she can convince herself that there is a one-to-one mapping of characters to graphical symbols. This system thus functions as a medium of experimental learning for the visitor. This project demonstrates an alternative way of designing user experience which (a) does not aim to foster among users an impression of mastery and knowledge, and (b) refuses to drive a wedge between the visual design and the underlying implementation. The graphical properties of the alphabet express the ASCII encoding of the characters, and so reveal the way in which the computer represents symbols.

The contents of the third screen, illustrated in Figure 6, closely resemble those of the second display, but with an important difference. The typographical symbols are constructed on the basis of Huffman codes, not ASCII codes. Since every symbol in this alphabet has seven possible strokes, and since the Huffman scheme reduces the length of the codes relative to the standard seven-bit ASCII scheme, every symbol typically encodes more than one character. As well, the code for one single character may be represented across two symbols (for instance, it may comprise the last stroke in the first symbol and the first three strokes of the second symbol). The mapping of characters to symbols is not one-to-one. Since the Huffman codes are continuously modified in response to every new key pressed, it is difficult for visitors to learn the current encoding scheme. The dictionary cannot be inferred from the visual feedback, because the dictionary is in a state of flux. The same graphical symbols often express very different codes. Since characters lack a fixed-length encoding, and since the visitor has no access to the underlying Huffman dictionary, it is impossible for her to know precisely how many strokes represent one single character. There is no way for visitors to map the characters they have typed to the symbols on the screen.
[Figure 6]

The content of the fourth screen differs strongly from the previous two (Figures 7 and 8).

[Figure 7]
My objective in this section of *Kinematograph* is to generate random characters whose probabilities reflect the frequencies with which those characters have been typed during the exhibition. The frequency data is of course stored in the Huffman dictionary. Throughout the exhibition period, the system continuously generates characters by making a sequence of yes/no decisions based on the probabilities associated with the nodes of the Huffman dictionary.\(^3\) The characters generated are

---

\(^3\) A short technical explanation is required to understand this display. Most programming environments have a random number generator. In this installation, I use the `random()` function built into Processing language, which produces random floating point numbers with a uniform probability distribution. I assume that the software represents every node in the Huffman tree as an integer. The variable \(j\) is first initialized with the value of the root node. I use an algorithm that, first of all, uses the standard `random()` function to generate a floating point number uniformly on the interval \([0, 1]\). I now use this number to select a character (one of the terminal nodes of the Huffman dictionary). The procedure that accomplishes this is as follows:

```c
int j = root;
float ran = random( 0, 1 );
while (j > max_value) {
    if (ran < probability(j))
        j = left[j ];
    else
        j = right[j ];
}
return (char) j ;
```

In this program, \(max\_value\) denotes an integer constant that stores the maximum possible ASCII code (i.e., 127). I assume that the Huffman dictionary has been constructed so that non-terminal nodes are assigned numbers greater than \(max\_value\), so that the while loop therefore continues until a terminal node is reached and the character associated with this node is then returned. I also presuppose the existence of some function `probability()`, which takes one parameter representing a node in the Huffman dictionary and returns the appropriate cumulative probability. If this cumulative probability is greater than the random number, the left child of the current node is selected; otherwise the right node is selected. (Here, left and right denote integer arrays that store the left and right children of every
not, however, displayed to the user. Instead, the system visualizes the operation of
the generative algorithm as an animated line: a horizontal stroke represents the
selection of a left node in the Huffman tree, and a vertical stroke represents the
selection of a right node.

Although this section of the system does not directly respond to user inputs, the user
nonetheless has an indirect impact on the animation. Since the algorithm relies on
the current Huffman dictionary, which is constantly being updated in response to
keyboard inputs from the visitors, the visualization depends on the accumulated
choices made by visitors over the course of the exhibition. They can affect the
visualization indirectly, by typing some characters more frequently than others.

The fifth screen demonstrates another way of thinking about the generative algorithm
just described. Every iteration in the generative process that produces the fourth
display involves a progressive elimination of possibilities. The algorithm traverses the
Huffman tree. At every node, a binary decision is made: whether to go left or right. If
the right node is selected at any given step, then all characters reachable from the
left node are thereby ruled out, and vice versa. This procedure is repeated until a
character is selected. Now, the fifth computer receives data from the fourth computer
and uses this data to visualize the gradual reduction of possibilities that takes place
during the generative algorithm. Every time the generative process begins anew, the
system draws a row of horizontal line segments running across the full length of the
screen. There are as many segments as possible characters (i.e., 128). The first
segment from the left represents the first possible character; the second segment
corresponds to the second character; and so on. As the system calculates further
iterations, new rows of segments are rapidly drawn below the first row, but the
segments corresponding to characters eliminated in each iteration are not drawn
(Figure 9). Like the fourth display, this process continues without interruption, even in
the absence of visitors.
Taken together, the fourth and fifth displays show two different aspects of the generative algorithm: (a) the algorithm consists of a sequence of binary decisions, such that (b) every decision eliminates a set of possible choices.

**Conclusions**

The installation never displays the characters typed by the visitor in their standard alphabetical form. Moreover, the various displays are updated even when the visitor types a key that does not represent any printable character, such as for instance the BACKSPACE. *Kinematograph* thus embraces the elimination of meaning that lies at the heart of information theory, and explores alternative ways of organizing thought and experience in this informational ecology. It raises the question of what it means to think, what it means to read and to write, with computational technologies that have no built-in semantics.

At the heart of this project is a fundamental reconfiguration of the problematic of information art. The concepts of “interactivity” and “interface” have often been presented as essential features of new media art. Interactivity is normally understood to require clear and immediate feedback to user inputs. Interfaces are often designed in line with principles of direct user manipulation. Systems are designed to be responsive and unobtrusive. Computation is thus conceptualized as a tool that augments our capacity to act. From this standpoint, the telos of informational technology is mastery. While *Kinematograph* continually responds to user inputs by updating the various visual displays, its principal aim is not user empowerment. Since there are no tasks for users to accomplish, its organization cannot be understood in instrumental terms.

The installation thus brings into being a way of relating oneself to technology that is not based on the desire for mastery. Technology is not essentially a means to an end. Instead, the system presents a situation of strangeness that facilitates a space for questioning. This strangeness primarily arises because it reveals certain technical features of information technologies, mainly the Huffman encoding algorithm, normally hidden from the user. The installation uses this technique to emphasize the dispersion of writing into multiple forms, and to resist all efforts to achieve purposeful mastery.

The relationship between inputs (the characters typed by the visitors on the keyboard), visual outputs (the five graphical displays), and the underlying algorithm at once invites exploration yet remains obstinately obscure. The installation invites visitors to participate while affirming its own self-autonomy and integrity. First of all, because those displays that respond immediately to user inputs subvert any expectation of clarity and control. The relevance of the system’s feedback to the user’s actions is not immediately evident. Secondly, because some of the displays (the fourth and fifth) continue to do their own thing even in the absence of any visitors. People cannot personalize and appropriate the system, which confronts them as an
alien object.

Whereas the standard ideology of interface design reinforces the role of the user who chooses certain inputs to produce expected outputs, the *Kinematograph* presents an alternative design model. This model affords an experience of engagement and estrangement, interactivity and alienation. The system promises to interact with the visitor but its feedback is not transformed into immediately comprehensible metaphors. This twofold experience dramatizes the fundamental alienation of the person confronting technologies that have become detached from normal human contexts, and which follow a purely formalistic logic.

References