

Filmes executáveis: da existência e propriedade das imagens em rede

Executable movies: on the existence and propriety of networked images

■ GABRIEL MENOTTI*

Universidade Federal do Espírito Santo, Departamento de Comunicação Social. Vitória-ES, Brasil

RESUMO

Este artigo explora a existência e propriedade das imagens em movimento em meio às redes de computador. Ele pretende demonstrar que imagens baseadas em computação são executáveis, representando processos de software da mesma forma que os aplicativos utilizados para reproduzi-las, editá-las, copiá-las e transmiti-las. Nessas condições, o filme e o dispositivo se tornam completamente misturados, desenvolvendo uma materialidade complexa. Evocando o conceito de código, a conclusão irá delinear como diferentes meios podem manter especificidades ainda que se tornem efeitos de superfície equivalentes do mesmo sistema sociotécnico.

Palavras-chave: Imagem técnica, software studies, cinema, materialidade, especificidades do meio

ABSTRACT

This paper explores the existence and propriety of moving images within computer networks. It means to show that computer-based images are executable, standing for running software systems as much as the applications that are used to play, edit, copy and transmit them. Thus, in computation, movie and apparatus become completely mingled together, developing a complex materiality. Analysing how computers enact and distribute moving images, I end up evoking the concept of code as a parameter that sets conditions for medial property. According to this parameter, different media might retain their specific identities even if they become equivalent surface effects of the same socio-technical system.

Keywords: Technical image, software studies, cinema, materiality, medium specificity

* Crítico e curador independente. Possui um doutorado em Media and Communications pelo Goldsmiths College, University of London (2011) e outro em Comunicação e Semiótica pela Pontifícia Universidade Católica de São Paulo (2012). Já trabalhou como visiting lecturer na Middlesex University e no Goldsmiths College. Atualmente é professor Adjunto do Departamento de Comunicação Social da UFES. E-mail: gabriel.menotti@gmail.com

A PROGRESSIVA DIGITALIZAÇÃO DOS canais de informação parece nos encaminhar para um mundo onde não mais restariam diferenças ontológicas entre sistemas midiáticos. De acordo com teóricos tão distintos quanto Friedrich Kittler (1999) e Lev Manovich (2008), na presente configuração tecnológica, tais sistemas são reduzidos a camadas de software rodando em redes de computador, que por sua vez assumem o papel de “máquinas midiáticas universais” (Manovich, 2008: 30). Esse processo de convergência, entendido em termos de uma desmaterialização dos suportes midiáticos, parece particularmente problemático para a organização do cinema, um meio cujas características essenciais foram tradicionalmente baseadas no uso de película de celuloide para a inscrição de imagens em movimento. Conforme o filme se torna obsoleto, e vários tipos de imagem passam a circular primariamente como padrões de dados binários, ainda seria possível diferenciar o cinema de outros sistemas visuais?

De acordo com o estudioso de cinema D. N. Rodowick, sim. Ponderando sobre “o desaparecimento da ontologia fotográfica” (2007: i), Rodowick declarou que “enquanto o filme desaparece, o cinema persiste” (Ibid.). Para ele, é o *jovem campo* dos estudos cinematográficos que deve se transformar, de maneira a dar conta dos modos como o tradicional meio subsiste na presente ecologia midiática. Ao abordar a infraestrutura do cinema tal como o aparato universal preconizado por estudiosos como Jean-Louis Baudry e Christian Metz, seus especialistas deixam passar a complexidade técnica das redes de computador. Como consequência, permanecem míopes às formas como tais redes podem expressar o objeto cinematográfico.

De modo a manter uma apreensão crítica desse objeto, Gertrude Koch sugere que os estudos de cinema poderiam ou se transformar ou se fundir a outros campos de estudos midiáticos (2009). Esse artigo dá um passo nesta última direção, propondo conexões dos estudos de cinema com disciplinas ainda mais jovens, e mais atentas à materialidade dos processos de armazenamento e transmissão de imagens, tais como o campo de estudos de software. Esse campo emergente se ocupa das interações entre software e cultura que sustentam “os novos meios de representação e comunicação” (Manovich, 2008: 4), sendo portanto, centrais na constituição do cinema após a computação. A partir daí, o artigo busca explorar como imagens em movimento são acionadas e distribuídas em redes digitais. Com isso, se pretende demonstrar que filmes baseados em computador são executáveis, representando algoritmos em funcionamento tanto quanto os aplicativos que são normalmente utilizados para reproduzi-los, editá-los, copiá-los e transmiti-los.

Tal demonstração será realizada por meio de uma análise da mecânica de sistemas de computação, suplementada por um exame detalhado de *And Then There Was Salsa* (2010)¹, uma obra cujo relato proporciona expressões exemplares das qualidades midiáticas do computador. Sem necessariamente focar nas particularidades de seu código, pretendemos explicar como sua natureza enquanto software mistura a constituição da imagem aos processos da máquina, conjugando-a a complexos vetores sociotécnicos e econômicos. Acredita-se que essa combinação metodológica permitirá uma compreensão mais apurada não apenas da natureza dos filmes digitais, mas também dos modos pelos quais as especificidades do cinema são gerenciadas após a convergência de mídias, ao mesmo tempo em que se mobilizam por novas camadas de operação e controle.

PRINCÍPIOS DA COMPUTAÇÃO: PRIMEIROS COMPUTADORES E ABSTRAÇÕES ALGORÍTMICAS

Para entender a natureza das imagens em movimento que resultam da computação, precisamos primeiro romper com a principal ilusão mantida pelas tecnologias digitais: a de que os meios se desmaterializaram. Como colocado por Matthew Kirschenbaum, “não existe computação sem que haja representação dos dados em um substrato físico correspondente” (2009: 27). Isso pode ser claramente percebido em máquinas de calcular precursoras do computador digital, tal como o ábaco. Datando do terceiro milênio antes de Cristo, esse equipamento organizava o cálculo conforme o movimento de contas em uma grade pré-definida. Tal correspondência entre operações físicas e abstratas foi perpetuada por toda história dos mecanismos de computação até a Máquina Universal de Turing, que N. Katherine Hayles caracteriza como “a base teórica para os computadores modernos” (2005: 176). O matemático Alan Turing mencionou esse dispositivo pela primeira vez em um artigo chamado *On Computable Numbers, with an Application to the Entscheidungsproblem*, publicado em 1936. No texto, Turing define os números computáveis como aqueles cuja “[representação] decimal pode ser inscrita por uma máquina” (1936), estabelecendo o registrado em suporte físico como condição de sua existência. Ademais, o artigo descreve uma máquina universal capaz de “computar qualquer sequência computável” (Ibid.). Apesar de hipotético, tal mecanismo parece inteiramente físico, consistindo de um scanner alimentado com um suprimento infinito de fita “análoga a papel” (Ibid.). A computação ocorre conforme a fita desliza pela máquina, que lê e anota símbolos nos segmentos discretos.

Alguns anos depois, trabalhando de maneira independente, o engenheiro alemão Konrad Zuse encontraria uma solução técnica similar ao modelo de Turing que lhe permitiria criar a primeira máquina computadora programável

1. Vinheta de propaganda criada pela Goodby, Silverstein & Partners para uma marca de tortillas da Frito-Lay e publicada no website da Vimeo (fev. 2010), popular plataforma de hospedagem de vídeos on-line.

completamente automática, o Z₃ (1941). Os programas desse dispositivo eram registrados numa espécie de fita perfurada muito particular: filme 35mm reciclado. Manovich vê esse fato como altamente simbólico: para ele, representa uma redução do meio “a sua condição original de veículo de informação, nada mais, nada menos” (2001: 25). Entretanto, é mais provável que as razões de Zuse para adotar o filme estivessem relacionadas às *affordances* e à disponibilidade desse material. Como foi apontado por Andrés Burbano, o filme possui certas qualidades físicas que favorecem a mecânica de computação. As perfurações e a divisão de quadros garantem uma alta precisão nos movimentos discretos que o cálculo digital requer. Além disso, o filme pode ser curvado, “criando ‘loops’ que permitiriam à máquina realizar operações recursivas” (Burbano, 2009: 9). Um último, mas talvez mais importante detalhe, é que Zuse tinha fácil acesso a esse material, já que seu avô trabalhava na indústria cinematográfica alemã (Ibid.: 7).

A arquitetura e o modo de funcionamento do Z₃ demonstram claramente que não existe nenhum tipo de *dataset* imaterial. Assim como os computadores digitais dos dias atuais, a máquina de Zuse operava de acordo com um sistema numérico binário, o que significa que os símbolos que empregava no processamento e armazenagem de dados eram apenas zeros e uns. Entretanto, esses *símbolos* não se tratavam da pura representação de valores abstratos que o computador *lia* – eles eram um arranjo de estruturas físicas, tão constitutivas do mecanismo de computação quanto as rodas dentadas o são de um sistema de engrenagens. Os zeros e uns na verdade se referiam à (1) presença ou ausência de furos no filme, que induziam mecanicamente (2) a posição dos interruptores que constituíam o processador da máquina, que por sua vez definiam (3) o estado das lâmpadas utilizadas como interface de saída do computador.

No Z₃, bem como em outras das primeiras máquinas de cálculo, esses arranjos eram interpretados como bits e então traduzidos em operações e valores matemáticos por especialistas humanos, chamados eles próprios de “computadores” (Hayles, 2005: 1). Entretanto, conforme as tecnologias eletrônicas se desenvolveram e cresceram em complexidade, a interpretação de dados pelo computador foi integrada às suas interfaces de entrada e saída. Esses processos de tradução foram internalizados em camadas de abstração de *software*, que codificam e decodificam padrões binários antes de qualquer operação humana, traduzindo os estados da máquina em formas mais próximas dos nossos sistemas simbólicos cotidianos (Ibid.: 108). Em outras palavras, essas camadas expressam a organização física da máquina como informações que um operador humano possa entender (como números, gráficos e textos). Como exemplos primários de abstrações de software, podemos citar as diferentes

linguagens de programação (que buscam aproximar a sintaxe da máquina daquela dos idiomas humanos) e as interfaces com o usuário (que buscam representar dados de acordo com metáforas instrumentais). Casos mais especializados seriam aplicativos com propósito definido, tais como *browsers* e programas de edição de vídeo.

Abstrações são necessárias porque elas preenchem a lacuna que existe entre os imperscrutáveis procedimentos do computador (e.g., acender uma complexa sequência de LEDs na tela) e os seus usos midiáticos (*rodar um filme*). Entretanto, ao fazê-lo, as abstrações também afastam a operação da máquina dos atuais processos de computação. Como disse Kittler, o software esconde a máquina dos seus usuários (1995); ele faz com que os usuários ignorem a própria fisicalidade do computador, juntamente com suas particularidades cinéticas e visuais, e restrinjam sua compreensão às representações metafóricas da máquina – como se o computador fosse efetivamente um *desktop* dinâmico, repleto de janelas redimensionáveis e ícones reluzentes, *ao invés de* um sistema para o processamento de informações.

A incorporação de abstrações cada vez mais sofisticadas aos sistemas de computação, aliada ao crescimento exponencial da capacidade de armazenamento e da velocidade de transmissão de dados (Kirschenbaum, 2009: 34), são as principais razões pelas quais a mídia computadorizada se comporta como se fosse imaterial. Não obstante, mesmo o computador mais moderno não passa de um mecanismo não muito diferente do Z3 – ou até mesmo de um ábaco. Apesar de suas diferenças físicas, todos esses dispositivos operam de acordo com princípios semelhantes.

Que princípios seriam esses? Em primeiro lugar, a computação implica certas formas de fluxo organizado. Embora isso seja óbvio em um ábaco, no qual o deslizamento de contas é uma operação nítida, é ainda mais prevalente num computador digital, em que até mesmo dados estáveis dependem de movimento. Os padrões binários não são armazenados em discos rígidos como rastros eletromagnéticos fixos, de um modo imediatamente apreensível, tal como os buracos de uma fita perfurada. Eles são alterações de voltagem na corrente elétrica que flui através desse aparelho, e, portanto, só existem quando o computador está em operação. De acordo com a descrição de Matthew Kirschenbaum,

o cabeçote de escrita/leitura [do disco rígido] mede a inversão entre campos magnéticos ao invés da mudança de um dipolo magnético individual. Em outras palavras, [o disco rígido] é um dispositivo diferencial – a significação depende de uma mudança no valor do sinal sendo recebido, e não da própria substância do sinal (2009: 90).

Outro princípio fundamental que pode ser inferido a partir da estrutura do computador é que os padrões de informação representam não apenas dados como também instruções para o seu processamento. Na máquina de Turing, por exemplo, os símbolos que eram inscritos em fita representavam não apenas os valores de entrada e saída, como também coordenavam todos os procedimentos formais de cálculo. Em computadores digitais mais complexos, está implícito que “todas as operações de código” – todas as abstrações de software – sempre se reduzem a uma mesma coisa: os supracitados “significantes de diferença de voltagem” (Kittler, 1995). Assim, meios computadorizados não admitiriam uma divisão ontológica estrita nem entre inscrição e transmissão, nem entre dados e instruções: no computador, tudo se reduz à contínua informação do mecanismo. Esse fato pode ser resumido na ideia de Kittler de que “não existe software”. Trocando em miúdos, software não seria *algo sendo executado em um computador*, mas sim *o próprio computador sendo executado de uma determinada forma*.

O que esses princípios de operação nos dizem a respeito do modo como os computadores acionam imagens em movimento? Primeiro, a equivalência essencial entre dados e instruções nos permite dizer que o arquivo de um filme constitui *software*. Mais precisamente, seria um padrão de informação abstrata: uma série de procedimentos formais que organizam o mecanismo computacional de acordo com sistemas de representação humanos (Kittler, 1995). Como tal, o arquivo do filme é insuficiente, uma vez que não é capaz de coordenar sozinho a operação de todo o sistema. Ele precisa interagir com outros algoritmos (como o sistema operacional e o aplicativo reprodutor de vídeo) para produzir imagens. Isso significa que o filme *não está contido no arquivo do filme*; o filme resulta do modo como o computador opera conforme ele incorpora e traduz as instruções parcialmente contidas no arquivo do filme. Essencialmente, o arquivo é apenas uma coleção de dados binários codificados, que não *significa* nada. Se o sistema não sabe as *regras* necessárias para acionar esses dados, ele não será capaz de formar imagens a partir do arquivo.

No que consistem tais regras? Mais do que o uso de um aplicativo apropriado, a reprodução de filmes digitais requer os codecs corretos. Codecs são algoritmos utilizados para codificar e decodificar informações audiovisuais: “Eles dimensionam, reorganizam, decompõem e reconstituem imagens e sons perceptíveis de modo que eles possam passar por redes de informação e meios eletrônicos” (Mackenzie, 2008: 48). Ao estabelecer um paradigma universal para a compressão de informação, os codecs reduzem drasticamente o tamanho de um arquivo de filme, permitindo o seu armazenamento e transmissão de maneira eficiente por meios digitais. Nesse sentido, J. D. Lasica aponta que é somente por causa do codec MPEG-2 que os DVDs se tornaram uma mídia

viável para filmes. Esse codec reduz em 97% a quantidade de dados necessários para o armazenamento de informações sobre imagens em movimento, fazendo caber um filme de longa metragem inteiro em um desses discos versáteis de 4.7 gigabytes, sem que haja perda significativa de qualidade (Lasica, 2005: 88).

Poderíamos dizer que os codecs estabelecem standards de organização de dados, tornando possível acionar as mesmas informações audiovisuais nos equipamentos mais diversos. Um filme codificado no popular formato h.264 (padrão de compressão de vídeo) pode ser exibido da mesma maneira em sistemas operacionais GNU/Linux, em media players portáteis *made in Taiwan*, nos smartphones da última geração, e no aparelho *Apple TV*, entre outros. Por outro lado, se o codec necessário não estiver instalado no sistema, o filme não poderá ser exibido de forma alguma (Mackenzie, 2008: 48). Logo, o codec não apenas “influencia a própria textura, ritmo e materialidade dos sons e imagens” (Ibid.), como representa uma parte constitutiva dos visuais exibidos, tão importante quanto o arquivo de dados do filme. Poderíamos até dizer que, ao assistirmos a um filme no computador, o que estamos observando é o trabalho dos codecs de desempacotar e organizar dados binários de acordo com a sua lógica espacial complexa (Ibid.: 51).

PERFORMANCE VISUAL DO SISTEMA E INTERAÇÕES COM O USUÁRIO

Para melhor entender a natureza das imagens produzidas pelas interações com computador, também precisamos considerar a visualidade inerente à própria máquina. É praticamente redundante apontar que as interfaces que enquadram e controlam a reprodução de um filme no computador são elas próprias software. Entretanto, quando levamos em conta a interdependência entre filme e dispositivo, esse fato passa a ter fortes implicações para a natureza do filme digital. Significa que essas interfaces não são armações fundamentais, tampouco rígidas: tanto quanto o filme digital, trata-se de organizações racionais da máquina, provocadas pelo modo como o computador processa informação. Partindo dessa premissa, a janela de um aplicativo *media player* não seria mais estrutural do que o filme sendo reproduzido *dentro* dela. Passando por cima de algumas sutilezas da arquitetura do computador, podemos dizer que tanto filme quanto dispositivo – incluindo aí o painel de controle que permite sua navegação randômica e até mesmo o sistema operacional em que o aplicativo está sendo executado – resultam da mesma interação de algoritmos. Todas essas *camadas* são produzidas concomitantemente enquanto o computador está rodando; tudo o que aparece na tela compartilha do mesmo caráter como abstração da máquina em tempo real.

Isso significa que o *filme* e as suas *interfaces* são efeitos de superfície equivalentes. Qualquer distinção ontológica entre a imagem que aparece dentro da janela de um aplicativo *media player* e aquelas que estão do lado de fora (como botões de controle e linha do tempo deslizante) não emerge da computação. Claro que são imagens distintas, mas não de maneira intrínseca. O que realmente separa uma da outra não é simplesmente que elas se comportem e reajam de maneiras diferentes às interações do usuário. (Em primeiro lugar, não são elas que estão reagindo, mas sim o computador.) Pelo contrário, o que separa uma imagem da outra é que o *usuário* reage de maneira diferente a cada uma delas. A separação entre *filme* e *interface* advém das expectativas que o público deposita na máquina, que vai conduzir o modo como ele percebe e se ocupa do sistema. Tais expectativas levam o usuário a ignorar, por exemplo, o fato de que o filme não seja mais espetacular do que o sistema em que ele está sendo reproduzido.

As interfaces com o usuário possuem qualidades visuais muito particulares, que bem poderiam ser caracterizadas como midiáticas. Inicialmente, isso pode ser difícil de perceber porque, na operação cotidiana do computador, o ato de observar serve principalmente como parâmetro para a agência reflexiva sobre o sistema. A dimensão *óptica* só importa enquanto se submete à dimensão *háptica*; o operador se encontra tão imerso na imagem que se torna difícil enxergá-la. Não é por acaso que as telas dos computadores se chamam *monitores*. Amiúde, a tela existe em função do mouse, dos joysticks, do teclado. A imagem frequentemente aparece para tornar possível a manipulação de conjuntos de dados, como um canal de entrada e saída.

Não obstante, quando nos distanciamos da operação da máquina, a pura visualidade do computador se faz evidente. Tal distanciamento pode ser histórico: aqui também pode ser útil nos voltarmos para os primeiros mecanismos de computação, de modo a notar como seus processos possuem uma dimensão visual muito marcante. No caso, poderíamos mencionar as imagens formadas acidentalmente pela inscrição de dados em cartões perfurados, bem como a animação dos tambores mecânicos utilizados pelo engenho analítico para efetuar cálculos². Como posto anteriormente, os primeiros operadores de computador precisavam interpretar esses padrões para convertê-los em dados. Hoje, a máquina interpreta os dados visualmente com antecedência, permitindo operações ainda mais complexas. Essa interpretação visual não se limita à mera transformação de valores binários em símbolos que os usuários possam manipular prontamente (números, palavras, imagens), como também gera uma série de coreografias que as interfaces desempenham de modo a se tornar mais amigáveis: janelas que criam suaves ondulações na tela conforme

2. O primeiro computador programável, concebido por Charles Babbage em 1837.

aparecem e dão cambalhotas ao serem minimizadas; menus de transparência ajustável, que jogam sombras uns sobre os outros; a área de trabalho pulsa com a previsão do clima e *lava lamps* simuladas.

Tais efeitos *eyecandy* não contribuem diretamente para a performance do sistema – isto é, para o volume de atividade útil que o computador pode realizar com os recursos de que dispõe. Em verdade, eles fazem o oposto, uma vez que consomem poder de processamento e memória que poderiam estar sendo empregados na manipulação *apropriada* de dados. Ainda assim, eles favorecem a operação do computador, porque proporcionam diferentes formas de feedback para os usuários, tornando a interação com a máquina mais orgânica. Logo, esses tipos de *performances visuais* do sistema existem para promover a sua transparência, tornando sua operação mais suave e dinâmica. É difícil atentar para tais performances justamente porque uma de suas principais funções é *produzir a invisibilidade* da máquina.

Pensemos por um instante nas qualidades plásticas do *Vimeo*, uma das mais populares plataformas de hospedagem de vídeos on-line. A interface de reprodução que ele utiliza é personalizada de modo a casar com a identidade visual do website, com campos de cores sólidas e formas geométricas de cantos arredondados. Quando uma página de vídeo é carregada, uma série de botões de controle surge sobrepondo a janela de exibição. Conforme o ponteiro do mouse passa sobre eles, os botões trocam de cor, como que indicando sua responsividade. Quando os botões são *pressionados* e o vídeo começa a rodar, essa interface desaparece em *fade*, desobstruindo a vista do filme. Essas animações sutis são produzidas pela operação do computador tanto quanto o vídeo em exibição dentro da janela. Não obstante, foram projetadas com o objetivo de negar sua própria visualidade. As animações fazem a interface parecer menos uma forma visual autônoma do que um aspecto da estrutura geral do website – menos *imagem em movimento*, mais *mecanismo*.

Apesar da impressão criada pelo design dos botões de controle do Vimeo, não existem separações fundamentais entre imagens e interfaces num sistema de computação. Isso pode ser demonstrado na peça *And Then There Was Salsa*, uma vinheta de propaganda criada pela Goodby, Silverstein & Partners para uma marca de molho de tortillas da Frito-Lay, publicada no website em fevereiro de 2010. Inicialmente, o vídeo parece se comportar como qualquer outro encontrado no Vimeo. Quando o usuário aperta o botão de *play*, o vídeo começa a rodar dentro da janela de exibição, como esperado. Trata-se da animação tridimensional de uma bailarina de flamenco dançando sobre colinas de legumes. Contudo, algo inusitado acontece. Conforme a personagem rodopia pelo cenário, a animação passa a escapar de seu *frame* habitual e a tomar conta

de toda a janela do *browser*. Primeiro, o layout da página explode, de modo que ela fica toda coberta de vegetação exuberante. Então, o frame do vídeo se alarga, enquanto tomates, cebolas e jalapeños voam ao seu redor (Fig. 1). Depois, a bailarina desliza através do fundo da página e, com um movimento certo, fatia o logotipo do site da mesma maneira que havia fatiado os legumes animados, revelando que uma imagem não é mais *cinematográfica* do que a outra. Assim, o que inicialmente parecia ser apenas uma interface operacional, contingente ao filme, se torna a parte mais interessante da obra.



FIGURA 1 – *And Then There Was Salsa* na plataforma Vimeo

And Then There Was Salsa literalmente mobiliza as características da interface de computador em favor do espetáculo visual. Desse modo, a obra coloca em questão a diferenciação entre imagens operacionais e espetaculares. Nessa linha, podemos especular se há realmente limites entre as possibilidades de interação com interfaces gráficas e a especiação de filmes digitais. O envolvimento com um sistema de computação regularmente envolve atividades como manipular o mouse, apertar botões metafóricos e prestar atenção em determinadas partes da tela. O modo como o usuário suprime essas atividades de sua consciência não parece tão diferente do modo como um espectador de cinema ignora sua própria situação na sala escura, em favor da experiência diegética. Em ambos os casos, é o envolvimento do público que abstrai certos processos técnicos da performance da máquina, enquanto torna outros relevantes para o valor e sentido da obra. Logo, embora o mecanismo do computador não separe espetáculo de operação, o público o faz, e dessa forma organiza a experiência da tecnologia midiática.

A CONSTITUIÇÃO DA IMAGEM NO COMPUTADOR

Além da falta de distinção fundamental entre filmes e interfaces, ainda existe outra característica importante dos filmes digitais que pode ser inferida a partir de sua existência como software: o fato de que dependem de um suprimento constante de informações do sistema. Em primeiro lugar, as imagens digitais nunca são realmente *inscritas*. Mesmo quando estão supostamente *armazenadas* em um disco rígido, elas existem apenas durante a transmissão de eletricidade por esse mecanismo (Kirschenbaum, 2009: 95). Além disso, uma vez que são armazenadas como conjuntos de dados *codificados*, elas só podem ser exibidas em decorrência de sua decodificação momentânea pela máquina. O que implica que mesmo as imagens mais estáticas, quando mostradas num computador, são consequência de interações procedurais – da atividade sem precedentes do sistema.

A atividade do sistema define o modo como os filmes serão acionados pelo processo de computação digital. Longe de serem simplesmente reproduzidos ou representados, os filmes literalmente resultam da operação da máquina. Essa é outra diferença sutil que pode ser mais bem explicada se nos referirmos a computadores mais antigos e simples. Vale uma analogia entre um PC moderno e a máquina de Turing. À primeira vista, podemos pensar que um filme sendo exibido no PC seria equivalente aos símbolos inscritos na fita da máquina de Turing, assim que ela termina seus cálculos. Entretanto, esse não é o caso. Tais símbolos seriam o resultado final da operação da máquina. Como demonstramos anteriormente, a imagem codificada não pode ser destacada da operação do computador dessa forma definitiva. Nesse sentido, seria mais esclarecedor comparar o filme que está sendo reproduzido na tela de um PC com a *fita* da máquina de Turing – mais precisamente, com os complexos padrões de movimento pelos quais ela passa, conforme é apagada e reescrita durante os processos de computação. Ao invés de tomar a imagem como um *produto* do computador, deveríamos considerá-la como seu índice em tempo real – insinuações da eletricidade que flui desde a fonte de energia, passa pela unidade de processamento e termina por excitar a malha de pixels da tela.

Antes de se tornar uma forma de representação, um filme digital é apenas o rastro fugaz do computador em execução. Eis um forte argumento contra o mito de que a imagem digital nunca degrada, e que pode ser copiada e reproduzida sem qualquer perda. Se aceitarmos a imagem digital como esse índice em tempo real, então precisamos admitir que ela *nunca é preservada* e que *não pode ser copiada de forma alguma*. Mas não porque a representação digital não possua “o molde espacial e temporal do evento originário”, que Rodowick defende como força causal primeira da imagem fotográfica (2007:

11). Tais representações possuem sim um evento originário – um do qual elas não podem ser desconectadas: os processos de computação. Ao mesmo tempo, é precisamente por causa desse vínculo material que as imagens digitais não podem ser estabilizadas. Afinal, o substrato necessário para computação não é um construto matemático infinitamente reproduzível, mas sim “um mundo bagunçado de matéria e metal” (Kirschenbaum, 2009: 27). Em outras palavras, imagens digitais são muito materiais para serem autônomas. Ainda menos que as imagens fotográficas ou eletrônicas, as imagens digitais não podem ser separadas de sua circulação.

Assim, se as tecnologias digitais “propagam uma ilusão de imaterialidade” (Ibid.: 135), é porque intensificam ainda mais a circulação de filmes, como fizera o vídeo anteriormente. A transmissão eletrônica havia promovido um regime dialógico que reduzia o intervalo entre a produção e o consumo da imagem. O processamento computacional, por sua vez, apaga a divisão entre as operações de inscrição e transmissão, fundindo o acionamento e a distribuição de formas. Ele intensifica o ritmo e a escala de circulação a tal ponto que a natureza do filme como um processo de informação se faz aparente à percepção humana.

O FILME ATRAVÉS DE REDES: MULTIPLICAÇÃO PONTO-A-PONTO DE FORMAS

Assim como o armazenamento e o mero *display* de uma imagem em uma máquina local, a cópia e transmissão de um filme por redes de computadores também envolve processos ativos de decodificação de dados. Para entender isso, podemos nos referir à operação de compartilhamento de arquivos ponto-a-ponto (p2p), um modelo de distribuição de dados que a indústria cinematográfica inicialmente associou à pirataria, mas que tem progressivamente adotado como um tipo de standard alternativo. Como apontado em outro artigo, uma das razões para a crescente popularidade desse paradigma é a eficácia com que emprega a estrutura dispersa da rede (Menotti, 2012). Em redes p2p, os usuários não baixam arquivos de um servidor central exclusivo, mas sim uns dos outros – por isso o nome. De modo a otimizar o tráfego pela rede, protocolos p2p podem assegurar que um único arquivo não venha inteiramente da mesma fonte. Isso significa que, quando um filme é obtido dessa forma, os dados são transferidos em partes desde diferentes lugares, sendo reunidos em um arquivo coerente apenas ao final do processo.

Levando em conta a dinâmica de operação distribuída do p2p, o artista Sven König e os membros do *!mediengruppebitnik* propõem uma analogia entre a operação dos codecs e aquela da transmissão de dados. Eles sugerem que, assim como o filme executado em um computador não está simplesmente contido

nos dados do arquivo, um *filme* encontrado em uma rede de compartilhamento não seria uma mera versão da obra original. Seria um trabalho completamente novo, que resultaria de um processo descrito como:

a soma do [>1] filme original, [>2] com o trabalho dos matemáticos que estabeleceram os fundamentos teóricos para [>3] os programadores que projetaram o software de codificação/ o codec e [>4] o usuário que finalmente emprega todos esse software para tornar o [>5] filme amplamente disponível. Os processos por trás de [2] - [4] normalmente permanecem invisíveis, causando a suposição errônea de que [1] = [5] (!mediengruppe & König, 2007).

Nesse sentido, podemos presumir que os processos de informação invisíveis que produzem as imagens computacionais não são apenas internos à máquina, como também ecológicos, dependendo da coordenação de transmissão e preservação de dados audiovisuais em uma infraestrutura mais ampla. Segundo essa perspectiva, os obstáculos para a completa autonomia da imagem são ainda maiores, a começar pela alta descartabilidade dos standards de armazenamento e codificação de dados (Usai, 1999: 44). Conforme o *milieu* técnico se transforma, os filmes digitais precisam ou ser traduzidos em novos formatos ou se tornar “hieróglifos”, como coloca o arquivista Paolo Cherchi Usai (Ibid.: 46). Por essa razão, Matthew Kirschenbaum descreve o eventual destino de todos os objetos digitais como

ser inexoravelmente reduzidos a blocos de código opacos, ou BLOBs, conforme eles se destacam e se afastam de seus ambientes de software nativos, e esses ambientes de software também se distanciam dos hardware em que rodam os sistemas operacionais que os suportam e sustentam (2009: 234).

Logo, a preservação de objetos digitais depende do acesso a sistemas capazes de hospedá-los e interpretá-los da maneira adequada (Kirschenbaum, 2009: 186, 189). Tal dependência ambiental pode ser explicada com nova referência a *And Then There Was Salsa*. A propaganda estava localizada em um website específico, e não poderia estar disponível em qualquer outro lugar. Sua ligação com o Vimeo ia além do espaço que ela ocupava nos servidores da plataforma e do modo como era promovida entre a comunidade de usuários do site. Para fazer os legumes voarem pelo browser, o filme teve que se apropriar da interface visual da plataforma e, principalmente, interagir diretamente com sua estrutura de software. Os realizadores tiveram não só que produzir a animação de acordo com a estética do Vimeo, mas também contatar seus webmasters com antecedência, de modo a entender como o site funciona e serem capazes de instalar algoritmos específicos nele. Certamente, todas essas definições demandam

negociações de longo prazo, e não é surpresa que a conta de usuário que possuía o vídeo ostentasse uma divisa de *patrocinador*.

No fim das contas, *And Then There Was Salsa* teve o mesmo destino trágico que têm outros objetos digitais, quando as condições ambientais necessárias se desfazem. O filme original não é mais encontrado on-line, tendo sido substituído por uma mensagem de erro que explica que ele foi apagado do Vimeo no dia 7 de Janeiro de 2011. Provavelmente, o contrato entre a agência de publicidade e o site expirou, tirando a página do ar e assim efetivamente destruindo a obra. Nesse sentido, a existência de um filme nas redes de computador parece extremamente frágil, uma vez que pode ser encerrada a qualquer momento, de acordo com a decisão dos gerentes da rede – uma deliberação que pode ser tomada por razões legais ou políticas, bem como empresariais.

Por outro lado, podemos entender que esses filmes subsistem precisamente ao serem deslocados e transformados. Contrariando a ideia de que a transmissão de objetos digitais seria um empreendimento puramente técnico, Kirschenbaum insiste em sua dimensão “fundamentalmente social”, cuja existência dependeria de práticas culturais particulares (2009: 21). O exemplo que ele cita é o da preservação de *Agrippa*, um texto eletrônico autodegenerativo, que se apaga ao ser lido. Antes que toda a restrita tiragem do texto sumisse por completo, hackers fizeram versões da obra para que fossem distribuídas em BBSs underground (Ibid.: 218)³. Nesse sentido, poderíamos argumentar que um filme digital é similarmente *preservado* ao ser re-codificado e multiplicado por diferentes canais, sendo neste processo convertido para diferentes formatos e resoluções. No caso, ainda que a conta de *patrocinador* que pela primeira vez publicou *And Then There Was Salsa* tenha sido suspensa, isso não significa que o filme tenha sido completamente apagado da Internet. Diversas versões ainda podem ser encontradas no Vimeo e em outras plataformas como o YouTube, publicadas por usuários que provavelmente não têm qualquer relação com seus realizadores originais. Algumas dessas versões são meras reproduções do vídeo, deficientes porque omitem as animações que passariam do lado de fora da janela de exibição. Outras, entretanto, são capturas de tela que mostram o vídeo no contexto de sua página animada, proporcionando uma documentação mais compreensiva do seu funcionamento (Fig. 2).

Tais multiplicações do filme bem poderiam ser vistas como seu desenvolvimento final enquanto um processo de informação, que entrelaça o valor e o sentido da obra no próprio tecido social do sistema, borrando as fronteiras entre produção e consumo midiáticos. É possível argumentar que tal condição de existência é promovida pela forma como o público está embutido na estrutura das redes de computador. Na internet, como numa brincadeira de telefone

3. Um acrônimo que significa Bulletin Board Systems, sistemas que permitiam às pessoas se conectarem aos computadores uns dos outros por meio de um terminal. BBSs foram populares na década de 1990, antes da Web.

sem fio, a audiência está efetivamente executando o processo de distribuição cinematográfica. Com essa transformação radical no modo como o público se relaciona com a estrutura do meio, o caráter da circulação do filme também se transforma, ressaltando a relevância da distribuição como uma espécie de acionamento coletivo da obra.



FIGURA 2 – Versão de *And Then There Was Salsa* na plataforma YouTube

Assim, as tecnologias digitais parecem embutir a obra ainda mais nos dispositivos cinematográficos, permitindo uma inversão: que as plataformas de distribuição penetrem a imagem e a utilizem para se propagar. Filmes obtidos em redes p2p, por exemplo, vêm frequentemente com um arquivo de texto listando o grupo de usuários responsável pelo upload original, bem como o diretório em que a obra foi postada pela primeira vez. Esses metadados servem para espalhar o nome de tais grupos e diretórios, garantindo sua influência na economia de reputação que move a cena do *filesharing* (Lasica, 2005: 53-55). A penetração de um site como o YouTube é ainda mais forte. À primeira vista, poderíamos achar que embutir a documentação de *And Then There Was Salsa* em outra página iria isolá-la de seu contexto original. Mas o que acontece é precisamente o contrário: o vídeo embutido vêm sobreposto com a marca e os banners do YouTube, bem como com links para outras obras em seu banco de dados. É como se, por meio do vídeo embutido, todo o YouTube viesse a infiltrar um outro website. Sendo assim tão invasiva, a plataforma reforça a sua autoridade superlativa sobre os trabalhos que hospeda – uma verdadeira forma

de controle, que se expressa de maneira mais clara na sua capacidade de banir usuários, tirar vídeos do ar e bloquear o acesso a partir de certos territórios geográficos.

O CÓDIGO E AS CONDIÇÕES DE PROPRIEDADE

A perspectiva dos estudos de software demonstra que os processos de computação misturam filme e dispositivo. Por um lado, os aplicativos digitais são operados por meio dos próprios efeitos visuais que eles produzem. Por outro, a imagem se dissolve completamente em seus procedimentos de transmissão e armazenagem. Em tais condições, o cinema só pode existir por meio da contínua interação entre mecanismos físicos e lógicos, tanto entre si quanto com operadores humanos, em plataformas locais e distribuídas. Ainda que isso não implique a desmaterialização do meio, certamente o torna muito mais volátil. Nesse sentido, com base em que fatores podemos insistir numa separação entre o cinema e as outras práticas midiáticas que dependem de tecnologia computacional?

Nesta última sessão, a resposta a essa pergunta será delineada evocando a noção de *código*. Caracterizada como um *standard* ou protocolo de comunicação, a codificação parece estabelecer condições para a especificidade midiática em um ambiente computacional. Nessa explanação será usada a abrangência histórica advogada por Friedrich Kittler, para quem códigos “não são uma peculiaridade da tecnologia computacional ou da engenharia genética” (2008: 40). Kittler encontra a origem da codificação em sistemas pré-cristãos de comunicação e comando que operavam por meio de criptografia. Ele caracteriza esse processo a partir da definição de Wolfgang Coy como “o mapeamento de um conjunto finito de símbolos de um alfabeto em uma sequência de sinais adequada” (Ibid.). Exemplos primários de tais sistemas são formas de escrita secreta tão antigas quanto o Império Romano, usadas tanto pelo governo quanto por conspiradores. Tais cifras são produzidas por meio da reorganização da mensagem de acordo com uma determinada *chave*, que opera como uma convenção particular dentro das convenções universais da linguagem. A circulação de uma mensagem criptografada é, portanto, restrita àqueles que compartilham de tal convenção. Desse modo, ainda que a cifra seja interceptada por algum agente inimigo, seu sentido não se revelaria. Logo, é possível dizer que a mensagem não está inteiramente *contida* na cifra – a mensagem ocorre quando a cifra é *operada* por meio da sua chave.

A cifra que supostamente contém os procedimentos algorítmicos de um aplicativo de computador é por vezes chamada de *código-fonte*. Essa alcuinha é enganosa porque dá a entender que o código seria a essência ou causa

fundamental do aplicativo. Pelo contrário, o código-fonte está sempre condicionado a uma série de chaves: as regras da arquitetura de programação em que o aplicativo deve ser executado. Logo, a verdadeira *fonte* de um aplicativo seriam as complexas interações de software que decifram seu código. De acordo com os princípios de computação descritos anteriormente, é preciso assumir que essas interações são elas próprias abstrações dos processos da máquina. Quando a operação do computador é reduzida a esses processos, nem mesmo os zeros e uns da linguagem de máquina existem de verdade – eles configuram uma mera descrição dos estados discretos do mecanismo em execução.

Sob essa luz, podemos começar a ver diferenças entre os modos como um conspirador romano e um computador pessoal *decifram* código. Enquanto o primeiro traduz o código em uma mensagem inteligível para então executar suas ordens, o segundo executa a mensagem em sua estrutura física, incorporando o código antes de traduzi-lo em efeitos inteligíveis. Isso significa que a máquina não entende o sentido de comandos como *goto* e *print*; ela apenas responde ao modo como esses comandos afetam seus mecanismos. Por esse motivo, a lógica algorítmica descrita em um código-fonte seria tão *superficial* quanto as imagens exibidas no monitor, no sentido de que também são abstrações das operações insondáveis da máquina. Parece então que o código não seria a causa fundamental das operações do computador, mas sim outra forma de racionalizar processos latentes na máquina. Colocando de outra maneira, digamos que a codificação não seria uma contingência genuína da computação, mas outro de seus efeitos de superfície.

Esse fato parece contradizer tudo o que expusemos até agora. Afinal, se um filme estiver codificado em um formato específico, ele não poderá ser executado em uma máquina que não possua o codec correto instalado. Logo, é possível dizer que o código define possibilidades fundamentais para a circulação de filmes. Acontece que mesmo essa suposição está ligada a uma noção particular do que é o filme e de quais são as formas apropriadas em que ele pode se manifestar. Contanto que haja recursos físicos, um arquivo de filme codificado em um formato específico *pode sim* rodar num computador que não possua instalado o codec correspondente; isso apenas não vai provocar a sequência de imagens esperada. Ainda assim, o computador irá processar o arquivo como informação. Mesmo que a máquina aparentemente *não faça nada*, isso não implica que não tenha executado milhões de cálculos. Por mais decepcionante que seja para o usuário, uma mensagem de erro é um efeito racional dos processos de computação. Ela também representa um modo como o computador exerce suas *affordances* materiais, da mesma forma quando um interruptor de teclado se fecha ou o processador fica superaquecido.

Assim, mais do que definir possibilidades fundamentais para a existência de um filme, o código especifica convenções por meio das quais os computadores podem ser utilizados de *maneira cinematográfica*. Estabelecendo tais convenções, o código organiza a tecnologia em favor do que seria a circulação apropriada do filme. Ao fazê-lo, a codificação cria uma camada de operação midiática, compreendendo um conjunto de práticas normais e standards informacionais de dados específicos. Empregando tais práticas, os usuários não precisam se preocupar em fechar os interruptores corretos ou acender pixels; eles podem simplesmente usar a máquina para criar e reproduzir filmes. De maneira similar, os standards de dados promovem a uniformidade das rotinas de renderização – isso é, do modo como as imagens são constituídas pelo mecanismo de diferentes computadores. Satisfazendo tais standards, um arquivo de filme pode ser manipulado de maneira semelhante pelos sistemas mais diversos.

Essa camada de operação midiática cria um terreno comum, no qual os filmes podem circular de acordo com as normas cinematográficas historicamente constituídas. Realizadores não precisam se preocupar com a computação; eles não precisam nem mesmo entender como funciona a codificação e decodificação de dados. Graças a esses standards codificados, estão livres para fazer cinema como sempre fizeram, empregando aplicativos cuja interface simula práticas estabelecidas de produção. Seu trabalho termina onde começa o dos codecs: no embrulho e desembrulho de bits em complexos arranjos semióticos.

Dessa forma, o código organiza computadores em favor da circulação de filmes, delineando plataformas comuns para a produção e consumo de obras cinematográficas. Ao coordenar a simulação de dispositivos e operações tradicionais do meio, o código define o que lhes é próprio. Qualquer coisa codificada como um filme, que seja portanto operável como um filme, passa a ser um filme. Possibilitando essa definição puramente arbitrária do objeto cinematográfico, as implicações do código são equivalentes àquelas de um *protocolo*. Alexander Galloway utiliza essa noção para descrever “como o controle existe após a descentralização”, numa rede distribuída que siga o modelo Deleuziano da sociedade do controle (Galloway, 2004: 29). Tal conceito também poderia ser utilizado para explicar como a especificidade do meio persiste num ambiente digital pós-midiático – em outras palavras, como ainda é possível identificar *cinema* quando o cinema não é diferente de outros bytes processador por computador.

Baseando-se na engenharia da computação, Galloway caracteriza protocolo como “um conjunto de recomendações e regras que definem standards técnicos específicos” (Ibid.: 6), que também estaria no cerne da computação em rede. É por adequar-se a um protocolo que os participantes se tornam capazes de se

conectarem uns aos outros e assim formar uma rede previamente inexistente. Nesse sentido, trata-se de uma tecnologia de inclusão (Galloway, 2004: 147). Entretanto, dada a maneira como está encurvadado no sistema, o protocolo também operaria como uma forma muito poderosa de *regulação*, sintetizando a negociação dos fluxos que estruturam a rede (Ibid.: 74-75). Estabelecendo condições de presença *de e em* um determinado sistema, o protocolo não criaria um campo de *discurso*, mas sim de *possibilidades* (Ibid.: 52-53). Uma vez que a mera existência na rede depende do cumprimento dessas condições, o protocolo não pode ser resistido: “opor-se ao protocolo é como opor-se à gravidade” (Ibid.: 147).

O objeto de pesquisa de Galloway é a própria internet, a rede global de computadores. De forma grosseira, podemos dizer que a internet opera como um convênio que simultaneamente distribui informação indiscriminadamente e regula essa distribuição por hierarquias de servidores. A liberdade de conexão universal depende da submissão ao controle protocolar. Não podemos enxergar tal controle como intrinsecamente nocivo. Sem ele, a rede perderia sua coerência: “Se a Internet fosse realmente rizomática, [...] ela resistiria aos usos profundos e significativos que as pessoas fazem dela cotidianamente” (Ibid.: 64). Assim, uma das principais funções do protocolo é reunir a arquitetura fragmentária da rede numa plataforma inteligível como a Web, que os usuários podem experimentar intuitivamente. De modo semelhante, por meio dos standards de dados, processos de computação são costurados em operações cinematográficas coesas, estabelecendo condições para a circulação de filmes no novo regime tecnológico.

Dessa forma, são estabelecidos parâmetros midiáticos para o envolvimento com sistemas digitais. Conforme os computadores se tornam o meio de todas as mídias, como profetizado por Kittler (2010: 225), esses parâmetros se tornam necessários para situar o cinema em relação a ele mesmo. Apontando para a pura especificação, o código define o que é próprio ao meio – o que fundamentalmente pertence ao seu circuito. Em meio às tecnologias digitais, o que não é codificado como um filme *não é um filme*. Ainda que possa parecer uma obra cinematográfica, o objeto enfrentaria vários impedimentos para circular como tal: o sistema operacional não o identificaria com o ícone apropriado; o leitor de DVD não seria capaz de reproduzi-lo; o festival de cinema não aceitaria sua inscrição.

Ainda mais problemático é o modo como tais dinâmicas de codificação misturam a questão *sobre o que pertence ao meio* com aquela *sobre a quem o meio pertence*. Conforme as tecnologias digitais tornam a existência do filme contingente aos codecs de vídeo e às plataformas de distribuição on-line, elas submetem o cinema às políticas destes bem como às patentes daqueles. Assim,

as redes de computador criam camadas que permitem maior controle legal e econômico sobre a mídia, mas que permanecem todavia escondidas sob a máscara de estruturas neutras. De modo a entender como essas camadas afetam o contínuo desenvolvimento do cinema, e como suas prerrogativas podem ser desafiadas ou deslocadas, precisamos fazer um esforço constante para alcançar além da tela, empregando métodos que iluminem as complexas propensões do progresso tecnológico. ■

REFERÊNCIAS

- IMEDDIENGRUPPE Bitnik & KÖNIG, Sven. what is the characteristic structure of p2p films? *Download Finished – FAQ*. 2007. Disponível em: <<http://www.download-finished.com/faq.html>>. Acesso em: 14 ago. 2011.
- BURBANO, Andrés. Between punched film and the first computers, the work of Konrad Zuse. *Re: Live Conference 2009*. Melbourne: The University of Melbourne & Victorian College of the Arts and Music, 2009.
- GALLOWAY, Alexander. *Protocol: How Control Exists After Decentralization*. Cambridge, Mass: MIT Press, 2004.
- HAYLES, N. Katherine. *My Mother was a Computer*. Chicago: University of Chicago Press, 2005.
- KIRSCHENBAUM, Matthew. *Mechanisms: New Media and Forensic Imagination*. Cambridge, Mass: MIT, 2009.
- KITTLER, Friedrich. There is no Software. *CTheory*, 18 out. 1995. Disponível em: <<http://goo.gl/saM3m>>. Acesso em: 14 ago. 2011.
- . *Gramophone, Film, Typewriter*. Stanford: Stanford University Press, 1999.
- . Code. In: FULLER, Matthew (ed.). *Software Studies: A Lexicon*. Cambridge, Mass: MIT Press, 2008. p. 40-47.
- . *Optical Media*. Oxford: Polity, 2010.
- KOCH, Gertrud. Carnivore or Chameleon: The Fate of Cinema Studies. *Critical Inquiry*, v. 35, n. 4, p. 918-928, 2009. DOI: <http://www.jstor.org/stable/10.1086/599582>
- LASICA, J. D. *Darknet: Hollywood's war against the Digital Generation*. Nova Jersey: John Wiley & Sons, 2005.
- MACKENZIE, Adrian. Codecs. In: FULLER, Matthew (ed.). *Software Studies: A Lexicon*. Cambridge, Mass: MIT Press, 2008. p. 48-54.
- MANOVICH, Lev. *The Language of New Media*. Cambridge, Mass: MIT Press, 2001.
- . *Software Takes Command*. 2008. Disponível em: <<http://goo.gl/D7fde>>. Acesso em: 14 ago. 2011.
- MENOTTI, Gabriel. Distribuição digital de películas: compromisso tecnológico, transgressão institucional y experiencia mediática em El estreno en línea de *Stealthis Film II*. *L'Atalante Revista de estúdios cinematográficos*. Espanha, n. 13, p. 12-19, 2012.

- RODOWICK, D. N. *The Virtual Life of Film*. Londres: Harvard University Press, 2007.
- TURING, Alan. On Computable Numbers, With an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*. Londres, n. 42 (1), 1936. DOI: <http://dx.doi.org/10.1112/plms/s2-42.1.230>
- USAI, Paolo Cherchi. A Model Image, IV. Decay Cinema: The Art and Aesthetics of Moving Image Destruction. *Stanford Humanities Review*. Stanford, n. 7.2, p. xiv-49, 1999.

Artigo recebido em 10 de setembro de 2013 e aprovado em 04 de novembro de 2014.