eikon journal on semiotics and culture

Received / Recebido
16 03 2017

Accepted / Aceite
03 04 2017

Author / Autor

Martina Annechiena Wieringa

Utrecht University
Netherlands

This is not the paint bucket you are looking for: the signs of Adobe Photoshop

Adobe Photoshop is among the most culturally influential computer programs. Like all software programs, signs and (interface) metaphors are used to convey meaning to the user. This essay discusses three Photoshop metaphors using Peircean, Saussurean, and Barthesian semiotics. The tool signs are motivated through remediative interface metaphors, seemingly rendering them icons. Yet, their object is code, making them indexical. Furthermore, these signs are simultaneously motivated and arbitrary.

Photoshop does not discriminate between remediated tools from different artistic practices; these tools are equal to one another, and interchangeable. Moreover, Adobe actively promotes a 'myth of artistry' by employing various dubiously grounded interface metaphors. The tools in Photoshop function as simulacra of the tenors they metaphorically allude to. The interface-metaphors are simulacra employed for the sake of familiarity and framing, but share no relation with the symbolically represented tools. User problems dealing with legibility, and usability, may be the result.

Keywords

Adobe Photoshop, semiotics, (interface) metaphors, myths, simulacra.

Interfaces communicate with their users. A typical graphical user interface conveys meaning through visual representation. These interfaces are constructed so that we, users, can (usually) understand their meaning and respond to them. The constructed nature of the graphical user interface (GUI) implies that the meaning it conveys is not 'natural'. Meaning – like the GUI itself – is constructed by people, through signs.

People agree that signs may mean one thing in one context, but something else in another. Signs are situated in signs-systems, which is the context in which their meaning is constructed, and agreed upon. For instance, the 'T' in physics has a different meaning than a 'T' in the Photoshop character window. While the first refers to temperature, the second is a showcase that displays the various font faces. These two signs thus, while they may look similar, do not share the same meaning due to their different contexts. This implies that meaning is something that has to be learned within the sign-system context (Chandler, 2007).

To help their users get acquainted with the sign-system context, interfaces often employ interface metaphors (Erickson, 1992). These metaphors tell us that something is like something else. For instance: a Photoshop canvas is like an actual world canvas, because it too can be used as a surface to be worked upon by an artist. Such interface metaphors thus tell us something about the possibilities of use of these various interface elements.

Metaphors, however, do not tell us something about the thing itself directly, but only do so by proxy (e.g. "it is a bit like…"). Thus, metaphors are at risk of becoming empty images, which do not actually correspond to the thing they are meant to explain. They are at risk of becoming simulacra: mere empty shells which may resemble something, but in fact share no characteristics with the thing they outwardly resemble. Simultaneously, the conscious deployment of the metaphors and simulacra may lead to a myth which surrounds the interface. For instance, the desktop metaphor, the recycle bin and the folder system in Windows are all metaphors which may lead to a myth of the computer office, because these metaphors alluded to office practices, such as filing, or objects, such as the desktop or the recycle bin.

Metaphors may thus be helpful, they may help us to contextualize certain features, or make a comparison between the software algorithms and the tools they are modeled after. Included in this notion of the helpful metaphor is a conception of a particular user: a user who needs guidance, or help in understanding the system. As a consequence, we risk getting 'locked in' in interface design: user needs change over time, with new generations, but the interface does not provide contemporary understandable models. Thus, as new generations of users grow up using systems which employ metaphors that do not hold any explanatory value to them, we run the risk of misunderstanding our software. In this essay I look at the way that Adobe Photoshop CS 5 Extended's interface deploys its signs, metaphors, simulacra and myths. I will do so by discussing three case-studies, in which a variety of mismatched metaphors are at play.

Photoshop's sign systems

Software – like Photoshop - is constructed to communicate the available variety of possible actions to its user. To aid the communication with the GUIs, language and images are used. To put it simply, the interface communicates through signs. These signs derive their value from the contrast with other signs within the sign-system (Gordon, 1996, p. 46).

To analyze the signs in Photoshop I will make use of semiology: 'the study of signs' (Chandler, 2007, p. 2). I will use both the three-part Peircean and the two-part Saussurian types of signs. The reason for this is that both Peirce and Saussure introduced concepts which are useful in understanding how Photoshop's interface works semiotically: motivated/unmotivated signs (Saussure) and the iconic, symbolic and indexical signs (Peirce). Another reason to use both of the conceptions is that Saussure has not incorporated a referent into the sign itself, while Peirce has. This may help us to perhaps uncover problems that are caused by the Peircean sign's referent, that do not exist for the Saussurean sign.

In Ferdinand de Saussure's book a sign is "anything that tells us about something other than itself" (Gordon, 1996, p. 14). He posed that a sign was made up of two parts: the signified, and the signifier. The signified is a "a mental representation of 'the thing'", while the signifier is a mediator to the interpreter of the sign, such as the written/spoken word or a visual representation of the thing (Barthes, 1967, pp. 42, 47). The signified and the signifier are inextricably connected and together they form the sign.

Charles Sanders Peirce had another conception of the sign. Both worked on their ideas separately, and while Saussure formulated the sign as a dyadic entity, Peirce regarded it as triadic (Chandler, 2007, p. 29). The three elements of Peirce's conception of the sign are the representamen (the form which the sign takes), an interpretant ("itself a sign in the mind of the interpreter" as Chandler (2007, p. 31) puts it) and an object (the referent). As in Saussure's model, all parts are essential for the sign to exist. This triad was based on yet another triad, which describes three different ontological modes: Firstness, Secondness, and Thirdness. Here, Firstness refers to that "which [something] is independently of anything else" (Houser, 2010, p. 90). Secondness deals with how it relates to other things. Thirdness, finally, deals with the "relations of relations, in a systematic arrangement" (Van den Boomen, 2014, p. 38).

The most important difference between the two conceptions of the sign is the presence/absence of the referent in the sign. While Saussure does not acknowledge a relation with actual world objects within his sign system, Peirce does. This is especially problematic for the signs in Photoshop, which often seem to refer to actual world objects, even though their relationship with them is obscure. In this paper, I will discuss the paint bucket tool, the burn tool and the sponge tool. Each of these case-studies highlights a particular issue with regards to the interface of Adobe Photoshop.

Paint bucket tool

A first case-study is the paint bucket tool in Photoshop. This tool uniformly colors a selected area, which is the signified. The button featuring the image of a paint bucket is used as a signifier to represent this function (Van den Boomen, Lammes, Lehmann, Raessens, & Schäfer, 2009, p. 274). Together they rather straightforwardly form a Saussurean sign.

When we turn to the Peircean sign we see the Saussurean signifier overlap with the representamen and the Sausurrean signified roughly overlap with the interpretant. What is new is the object: the thing outside the sign to which the sign refers. The addition of this component complicates the paint bucket tool sign. The object to which the paint bucket tool seems to point is an actual world paint bucket. Thus, the sign seems to be an icon. An icon – in the Peircean sense – resembles its object. This likeness is not

necessarily visual. An icon is something that "is like that thing and used as a sign of it" (Peirce, 1998, p. 291). Using a paint bucket, however, would not achieve the same effect as its digital counterpart. The actual paint bucket would lead to a more splattered, messy and uneven effect. Photoshop's paint bucket is much more clean, flat, precise and sterile compared to its object counterpart. This complicates the iconic nature of the sign, as it does not seem to resemble a paint bucket that much, yet it does seem to remind us of it in a more metaphorical way.
But while the representamen of the sign is a button which shows the image of a paint bucket, the name of the tool is 'paint bucket tool', it need not point to an object paint bucket per se. As Adobe Photoshop is a software program, all of its tools are essentially code. The object of the paint bucket tool would in fact be the code procedures/algorithms that allow the user to use the tool itself. In this sense the sign is indexical. An index – unlike the icon – does not represent the object so much, as is a sign of it because it is connected to it (Peirce, 1998, pp. 460–461). The index exists because its object exists, but – contrary to the object - its interpretant is not essential to its indexicality (Peirce, 1932, p. 304). An index points to the object, in this case the 'paint bucket tool' points to the code that allows the user to use the tool.

Yet the sign – though not iconic – is motivated: it leans on other signs. As a sign like this leans upon other signs, it is no longer an arbitrary sign. The motivation of naming it a 'paint bucket tool' and to use a paint bucket image for its button, seems to have its basis in remediation (Bolter & Grusin, 2002). By picking an already known object – which is associated with artistry, such as Pollock's action-painting - for the (Peircean) sign to refer to, Adobe points to older and known forms of artistic practice, in this case painting.
Photoshop is a digital medium, while a paint bucket refers to an 'analogue' practice. Thus, the paint bucket tool does not only claim to be artistic, it claims to be able to transpose the 'analogue' artistic practice to the digital realm as well: it remediates the artistic tool in code and in the interface metaphor. This transposition essentially frees the sign from any actual world object. While it seems iconic at first glance (and thus related to an actual world paint bucket), it is in fact a motivated, indexical interface metaphor, that refers to the underlying code (Ryan, 2002).

Burn tool

This play with remediation does not always work out the way it might have been intended by Adobe. In some cases younger people working with Photoshop might be unaware of the remediative motivation of the sign. Such is – for example - the case with the burn tool for the post-darkroom generation. In this particular case remediation can occur in an inverted way, when people who are not familiar (and are thus unable to acknowledge the remediative, metaphorical nature of the representamen) with darkroom procedures start acquainting themselves with this tool.
For these people the object of the sign is not recognizable, therefore it is arbitrary, perhaps doubly so, as the intention was to make it a motivated sign. What once was a motivated sign then suddenly becomes an unmotivated one. Instead of recognizing that the tool remediates its darkroom counterpart, the user is faced with an arbitrary signifier. Perhaps they are not even sure what it signifies, but as soon as the tool is used, the signified will reveal itself: darkening the area of choice. When the metaphor is illegible to the interpreter of the sign, the sign ceases to be motivated, as the motivation of the sign rests in its metaphorical nature. When the interpreter finds out about the intended motivation, the sign only further emphasizes its artificiality through this remediation attempt.
So why would Adobe continue to use the burning procedure as the object for the sign's representamen if it is not recognized by the interpreter/user and emphasizes the construction of the signs of the toolbar? When a sign is arbitrary it doesn't matter what signifier is used, really. It does not matter if the sign originally was motivated. The interpreter/user needs to figure out the signified regardless of the signifier if it is not legible at first sight. Here, the originally metaphorical relation is turned into a symbolic one. A symbol represents its object, not per se because it resembles the object or because of any real or obvious connection, rather the representation of a symbolic sign rests on habit and convention (Peirce, 1933, p. 531, 1998, pp. 460–461). The symbol is a sign because it is interpreted as such. The association 'darkening a selected area' and the burn tool representamen is based purely on habit (e.g. employing this tool results in that particular effect) instead of a metaphorical relationship with the darkroom procedure of burning that is noticed by its user.
Here, then, the metaphor is used to frame the software in a particular way – and if it fails to reach the newer generation, so be it. This is highly problematic, as it in fact hinders usability of the product. Not all graphic designers are the 'homo universalis' which Photoshop assumes them to be, which in this case hampers their understanding of the tool's function.

The referent of the burn tool is as problematic as the paint bucket tool's. Digital cameras no longer have photosensitive films, for which exposure is of key importance. Terms like 'burning' and 'dodging' which (metaphorically) refer to this exposure, no longer make any sense when the photograph is comprised of binary data and is not carried by photosensitive film. This does not mean that the procedures that 'burning' and 'dodging' represent are no longer meaningful, it just mean that us calling them that way is completely arbitrary and rests on remediative habit. The difference between the burn tool and the dodge tool is arbitrary as neither has to do with actual exposure, anymore. In fact, one could argue that they both do the same thing: manipulating the underlying data to simulate more/less exposure time. Yet, Photoshop remediates this difference by creating different buttons for each tool. The buttons are located in the same spot on the toolbox, and they cannot be on top/visible at the same time. The burn and dodge tool are equated (position wise) with the sponge tool, which I will discuss below. These tools are then interchangeable, if we go by the logic of the interface. One of these will be visible, but can be swapped for any of the two others. Thus, while the tools metaphorically allude to different disciplines and practices, they have become democratized in the Photoshop interface. They are indexical signs that point to interchangeable functions for the user to be employed in the practice of manipulating an image.

The organization of these tools is, however, not based on their remediative source material (we will see below that the 'sponge tool' – problematically - alludes to water color painting) but on similarity of the procedure of use. All three make use of brushes that designate the area that is to be affected. These brushes can take various forms. All three of the tools have options for the user to pick from: range (midtones/shadows/highlights) and ex-

posure for the burn/dodge tool, and mode (desaturation/saturation) and flow for the sponge tool.

Sponge tool

The sponge tool, like the paint bucket tool and the burn tool seems to refer to a 'sponge' at first glance. Sponges are used in watercolor painting to soak some of the paint up and thus 'desaturate' the painting (Johnson, 2009). In Photoshop, however, the sponge tool also allows the user to saturate the painting. Here the sponge tool, even though it tries to remediate an actual sponge, gets equipped with qualities that seem the exact opposite of what an actual sponge does (namely to soak something up).
As with the burning and dodging of digital photo's, which need no photochemical light exposure process, the sponge tool is an arbitrary metaphorical remnant of remediation. But where the burn and dodge tools would be logical remediative metaphors for those familiar with darkroom procedures the sponge tool is not as obvious for those who have practiced watercolor painting. What the burn and dodge tool do is, however fundamentally different, still similar to their referents. They emulate those practices, albeit in a 'refined' way because one can choose to affect, for example, only midtones. The sponge tool has a new function that does not correspond with the metaphor's vehicle: namely to saturate the painting. The sponge tool can not only be used to desaturate (which was its actual world sponge's sole function) it can also add more 'pigment'. Moreover, it can saturate colors/pigments that it has not been in contact with before, and can switch between saturation/desaturation mode instantly. An actual oversaturated sponge would merely leave a smudge of color and could never do so without first soaking pigment up.
The discrepancy between the metaphor and the sign is caused by the different things they refer to. The actual world object of the sponge tool sign, as we've seen before, is not a sponge, but the underlying code it refers to. This code is not materially hindered to saturate or colors like the actual world sponge would be. The difference for the sponge tool to desaturate or to saturate is a different setting that the user can opt for, which results in the deployment of a different piece of code. Because of the transfer from analogue to digital, the tools are not limited to the material constraints of the actual world practices they remediate. They are merely bound by the constraints that the code imposes on them.
The metaphor of the sponge tool, however, does point to an actual world sponge. The sign (and most notably its representamen/signifier) functions as the vehicle (the 'image'), while the actual world sponge is the tenor of the metaphor. The relation between vehicle and tenor is motivated, and this motivation lies in what the two have in common, which is usually referred to as their ground. In this case the ground of the sponge tool and an actual world sponge is that they can both desaturate the 'painting' (Van Boven & Dorleijn, 2010, p. 162). As an interface metaphor, this helps us to understand "the design or mode of operation of a computer application" (Ryan, 2002, p. 583). An interface metaphor, then, helps us to understand how the various tools in Photoshop can be employed, because it gives us a frame of reference for its usage.
Yet simultaneously the ground upon which this metaphor functions is more than dubious. Because the sponge tool is also able to do the precise inverse of an actual sponge, the sponge tool metaphor undermines its own ground. The interface metaphor seems to function as a simulacrum: an image freed from its ground (Deleuze, 1994, p. 272). The simulacrum of the sponge tool calls an actual sponge to mind, refers to it in its name, and metaphorically asserts that it works in a similar way. Yet it is fundamentally different from an actual sponge. It bears only an outwardly resemblance to the object, but does not actually correspond to the thing. It is not a copy of a sponge, but only a superficial remediative effect. One does not resemble the other. One is not a variation of the otherwise same other (Parr, 2010, pp. 74–75). They are singular phenomena, which only differ from each other. Thus, there is only difference (Deleuze, 1994, pp. 273, 299; Massumi, 1987), a difference which is freed from its resemblance and variations of sameness (Parr, 2010, pp. 74–75). The sponge tool interface metaphor, then, is a hollow simulacrum which points to nothing more than its fundamental otherness.

Photoshop's artistic myths and simulacra

The Photoshop user is able to fill a 'canvas' with 'paint' and use 'photography procedures', like the burn tool, or 'watercolor techniques', like the sponge tool, on it. The material that is represented on screen belongs to all yet neither of the various traditional artistic genres. It seems to combine various aspects of the different traditional materials, and goes beyond the material limits of these traditional tools: the code is the limit.
Even when the user explicitly imports a photograph, for example, he/she can still use the sponge tool to (de)saturate its colors. Once the material (be it a drawn image/photograph etc.) is imported into Photoshop, the material is democratized. It can be edited with procedures that refer to a variety of artistic practices that were previously seen as distinct. In the program these procedures can be used indiscriminately.

What we see in Photoshop, then, is code cloaked in remediative metaphors. These employed interface metaphors do, however, not cover the nature of the code completely. These metaphors help users to make abstract algorithms and code more concrete (Erickson, 1992, p. 66). The signs of the various tools are used to communicate the various possible actions to the user. Additionally, they allude to various artistic practices, but their object remains the code that governs the function/tool/procedure. This code democratizes all the various artistic genres that are alluded to in the interface metaphors, in the various signs. Photographic procedures, water coloring equipment and paint brushes are each other's equals on the code level. The interface has rendered them interchangeable.
Because of their transfer from the actual to digital world, the tools are not limited to the material constraints of the actual world practices they remediate. They are merely bound by the constraints that the code imposes on them. As a consequence, the referents of these tool-signs no longer point to an actual world tool or procedure. Instead, the image of these actual world objects is used to create an explicitly remediative metaphor that only partly corresponds to an original. The object of the sign (e.g. the paint bucket tool) is not the object we would expect it to be (e.g. a paint bucket), instead, the object refers to an obscured set of coding. This is something that goes against Peirce's view on metaphors as hypoicons', signs which are particularly iconical. Such hypoicons "represent the representative character of a representamen by representing a parallelism in something else" (Peirce, 1903). According to Peirce (1903), the object of an iconical sign needs to be a Firstness – here we see that this is no longer the case.
Moreover, the relation between the object-code and the tool-sign is, however remediatively motivated, also sym-

bolic. While the relation between the sign and the interface metaphor – which conveys a sense of the possibility space to the user - is motivated, the relation of the sign to the object-code is symbolic and arbitrary, because we can't infer the direct relationship between code and the perceived software tool. Any number of signs and metaphors could have been chosen to represent these particular lines of code, to serve as a symbolic index for them – as there is no iconic relationship between representamen and object.

The representamen, however, acts as an interface metaphor in its communication with the user. The arbitrary representamen functions as the vehicle of this metaphor and communicates the possibility space of the tool, the tenor. This metaphor is motivated by ground that the representamen shares with an actual paint bucket/burning procedure or a sponge (Van Boven & Dorleijn, 2010, p. 162). For instance, the sponge tool is like a sponge, because it can desaturate a colored surface. But, because the sponge tool can transcend the material limitations of the sponge, and is only hindered by the limitations of code, it has the extra capacity of saturating the image. This results in a ground that is dubious at best, as it denies its own similarity to the actual sponge in the act of transcending the metaphorical likeness. The relation between the two is not one of similarity, but one of difference and empty outwardly resemblance: it is a simulacrum. Yet it is a motivated simulacrum at that.

By favoring the terms like 'paint bucket tool', 'burn tool' or 'sponge tool' Adobe constructs a narrative in its interface. This is what I term the myth of artistry. It is no coincidence that these terms and metaphors are used in the program, rather, it is an intended construction by Adobe. As we've seen with the sponge tool, the sign itself is arbitrary – the motivation rests wholly on the metaphor it employs. These are metaphors which rest not on Firstness, but on Thirdness.

By employing these particular metaphors, Adobe seems to place its program in a canon of artistic tools. But while these digital tools allude to their analogue counterparts, they are not alike. They do not afford the same possibilities entirely, but they may correspond on some level. For example, the sponge tool – like the burn tool and the paint bucket tool – seems to be a more sophisticated piece of equipment than their actual world tenors. One does not risk smudging the rest of the picture when using the sponge tool, one can use it on one layer instead of the entire plane, the action is reversible and the tool has more options than its referent. Similarly, burning/dodging requires less precision of the user (as actions are reversible), and is less time and physically demanding (where the analog photochemical burning dodging processes were irreversible and both physically and time demanding). Similarly, if one were to try and cover a surface with an actual paint bucket, it would be a messy (and quite irreversible) affair, while in Photoshop its clean, sterile and reversible. Taking this into account, it becomes clear that Photoshop tries to frame itself as a more sophisticated or evolved tool for artists by direct comparison to older forms of art production. The remediative metaphors that are used in the interface are more precise, clean, reversible and often have more options than the actual world objects than are their tenors. Yet the metaphors are self-contradictory and thus expose themselves as simulacra. These simulacra do not all function in the same way. While they are arbitrary, interchangeable and democratized in nature, they metaphorically refer to different sign-systems (e.g. that of photography, painting, drawing and so forth). It is this eclectic, simulated agglomeration of signs that, when it is simulation is uncovered, loses its depth (Jameson, 1991, p. 34). The empty simulacra are exposed as being devoid of any meaning: they are only superficial images, mirages, and facades (Van den Braembussche, 2000, pp. 353–354).

Conclusion

The sign-system of Photoshop employs various signs to communicate to the user. In this essay I have discussed the paint bucket tool, the burn tool and the sponge tool. The signs of these tools are motivated through remediative interface metaphors, which gives them the feel of iconic signs. Yet, their object is not found amongst the corresponding traditional artistic tools, but in the code that governs the possibility space that is afforded by the tool in Adobe Photoshop. The tool-signs are motivated due to their external relationship to the user (who is grounded in an actual world to which the metaphors' tenors belong), yet are arbitrary at the same time.

The Photoshop tools are democratized by the interface, as most of them are interchangeable due to the interface organization. The burn tool, dodge tool and the sponge tool can never be displayed on top of the tool box at the same moment in time. Only one of these is able to be on top. The user has to swap one for the other. This democratization of tools leads to a sign-system in which the signs are equal. This equality, however, is problematic, as the various tenors of the interface metaphors originate in different artistic traditions and may share more or less ground with their vehicles. Within the interface then, the signs are interchangeable and thus rest on equality, while the metaphorical ontology of these signs is one of inequality. In a nutshell, while all of the signs point to their coding in the same way, not all of the metaphors share the same amount and sort of likenesses between the vehicle and the tenor.

Adobe actively promotes a myth of artistry by employing the various interface metaphors. By using these metaphors, Adobe seems to insinuate that it transposes analogue artistic practices in a digital environment. But this is not where the myth ends. All of the case-studies that were discussed in this essay were more 'sophisticated' than their tenor counterparts. The Photoshop paint bucket does not drip, it does not spill, it is sterile, contained and reversible. The burning tool does not require any lengthy and precise physical movement on the part of the user. All one needs to do is drag the cursor over the desired areas to affect the 'exposure'. Furthermore, this digital burning is more precise, as the user can opt to affect only a specific range of tones and is reversible, whereas the darkroom procedure would require to make a new print if the developer made a mistake. The sponge tool is a slightly different case. Whereas the paint bucket tool and the burn tool emulate their actual world tenors in a more precise and sophisticated way, the sponge tool adds something to its tenor that it could never do. The sponge tool can decrease and increase image saturation, whereas an actual world sponge is normally only able to desaturate the image. For the sponge tool, the material constraints of the tenor are replaced by the constraints of code, which now allows it to also perform the inverse of its tenor.

The ground upon which the metaphorical likeness is based is thus undermined. The metaphor seems to be founded on nothing but mere outwardly resemblance, emptied out of the characteristics the vehicle and the tenor were assumed to share. The tools in Photoshop function as simulacra of the tenors they metaphorically allude to. They are not mere 'copies' of their analogue tenors, they are of a different order entirely. These simulacra refer to different sign-systems (that of painting, water color painting, pho-

tography and so forth), by calling them to mind, remediating them in name and metaphorically allude to them. Yet, while they are thus motivated simulacra, they are without depth. The interface-metaphors are simulacra employed for the sake of familiarity and the frame of a canon of artistic practice, but share no relation with the tools they symbolically represent.

Referências bibliográficas

BARTHES, R. (1967). **Elements of semiology**. New York: Hill and Wang.

BOLTER, J. D., & Grusin, R. (2002). **Remediation** (Fifth). Cambridge: MIT Press.

CHANDLER, D. (2007). **Semiotics: the basics** (2nd ed.). Abington: Routledge.

DELEUZE, G. (1994). **Difference and repetition**. London: The Athlone Press.

ERICKSON, T. D. (1992). *Working with interface metaphors*. In B. Laurel (Ed.), **The art of human-computer interface design** (pp. 65–73). Reading: Addison Wesley.

GORDON, W. T. (1996). **Saussure for beginners**. London: Writers and Readers Limited.

HOUSER, N. (2010). *Peirce, phenomenology and semiotics*. In P. Cobley (Ed.), **The routledge companion to semiotics** (pp. 89–100). New York: Routledge.

JAMESON, F. (1991). **Postmodernism, or the cultural logic of late capitalism**. London/New york: Verso.

JOHNSON, B. (2009). **Earthbound Light - Dodge, Burn and Sponge the Photoshop CS4 way**. Retrieved from http://www.earthboundlight.com/phototips/photoshop-cs4-dodge-burn-sponge.html

MASSUMI, B. (1987). **Realer than Real: The Simulacrum According to Deleuze and Guattari**. Copyright, 1(1), 90–97. Retrieved from http://www.anu.edu.au/HRC/first_and_last/works/realer.htm

PARR, A. (Ed.). (2010). **The Deleuze Dictionary**. Edinburgh: Edinburgh University Press.

PEIRCE, C. S. (1903). **On Some Topics of Logic**. Boston.

PEIRCE, C. S. (1932). **Collected Papers of Charles Sanders Peirce. Vol. 2: Elements of logic**. (C. Hartshorne & P. Weiss, Eds.). Cambridge: Harvard University Press.

PEIRCE, C. S. (1933). **Collected Papers of Charles Sanders Peirce. Vol. 4: The simplest mathematics**. (C. Hartshorne & P. Weiss, Eds.). Cambridge: Harvard University Press.

PEIRCE, C. S. (1998). **The Essential Peirce, Selected Philosophical Writings** Volume 2 (1893-1913). (Peirce Edition Project, Ed.). Indiana University Press.

RYAN, M. (2002). **Beyond myth and metaphor: Narrative in Digital Media**. Poetics Today, 23(4), 581–609.

VAN BOVEN, E., & Dorleijn, G. (2010). **Literair mechaniek** (2nd ed.). Bussum: Uitgeverij Coutinho.

VAN DEN BOOMEN, M., Lammes, S., Lehmann, A.-S., Raessens, J., & Schäfer, M. T. (Eds.). (2009). **Digital Material: tracing new media in everyday life and technology**. Amsterdam: Amsterdam University Press.

VAN DEN BOOMEN, M. (2014). **Transcoding the Digital: How Metaphors Matter in New Media**. Amsterdam: Institute of Network Cultures.

VAN DEN BRAEMBUSSCHE, A. A. (2000). **Denken over kunst: een inleiding in de kunstfilosofie** (3rd ed.). Bussum: Coutinho.