ARTIFACT 2007 | Volume I, Issue 2 | Pages 76-84

Deep Remixability

by Lev Manovich, University of California, San Diego

This article is the second part of the series devoted to the analysis of the new hybrid visual language of moving images that emerged during the period 1993–1998 and which today dominates our visual culture. In the first part the author suggested that the new language can be understood with the help of the concept of remixability if we use this concept in a new way. We can call this "deep remixability", for what gets remixed is not only the content of different media but their fundamental techniques, working methods, and ways of representation and expression. In the earlier article discussion was started of how the new software-based methods of production – specifically software such as After Effects – made this language possible. In this part the author continues to explore different effects of software-based moving image production, and then uses this discussion to refine his analysis of how deep remixability functions. He discusses how today the design of moving image sequences takes place within three-dimensional space, with 3D compositing gradually replacing 2D compositing. He also looks at the shift from the concept "moving image" to a new concept of "media composition". Finally, he analyzes how the typical production workflow in a contemporary digital studio the ways in which a project moves from one software application to another – affects contemporary visual aesthetics of not only moving but also still images. The article shows that this workflow has two fundamental effects. On the one hand, never before have we witnessed such a variety of forms as today. On the other hand, exactly the same techniques, compositions, and iconography can now appear in any media.

Keywords: aesthetics, design, interface, film, motion graphics, software, visual culture

INTRODUCTION

This article is the second in a series devoted to the analysis of the new hybrid visual language of moving images that emerged during the period 19931998. Used first in film titles and television graphics, this language slowly came to dominate our visual culture. Today we see it in short films, music videos, commercials, moving image sequences that appear in interactive projects and media interfaces, and websites. Because of this fundamental shift in the aesthetics of moving images it did not receive any critical discussion while it was happening – in contrast to other aspects of the Digital Revolution such as interactivity and the Web – I have called it a "Velvet Revolution" in moving image culture.

My thesis is that this new language can be understood with the help of the concept of remixability – if we use this concept in a new way. Let us call it "deep remixability", for what gets remixed is not only the content of different media but their fundamental techniques, working methods, and ways of representation and expression. United within the common software environment, cinematography, animation, computer animation, special effects, graphic design, and typography have come to form a new metamedium. Awork produced in this new metamedium can use all techniques that were previously unique to these different media, or any subset of these techniques.

In the first part, entitled *After Effects, or Velvet Revolution*, I started the discussion of how the new software-based methods of production – specifically software such as After Effects – made this language possible. We analyzed compositing; we also discussed how the interface and production workflow in After Effects themselves mix the production methods of twentieth-century cinema, animation, and graphic design. In this part I will look at other aspects of software-based moving image production, and then use this discussion to refine my analysis of how *deep remixability* functions.

THREE-DIMENSIONAL SPACE AS A NEW PLATFORM FOR MEDIA DESIGN

As I was researching what the users and industry reviewers had been saying about After Effects, I came across a somewhat condescending characterization of this software as "Photoshop with keyframes". I think that this characterization is actually quite useful.[1] Think about all the different ways of manipulating images available in Photoshop and the degree of control provided by its multiple tools. Think also about its concept: a visual composition as a stack of potentially hundreds of layers, each with its own transparency and multiple alpha channels. The ability to animate such a composition and continue using Photoshop tools to adjust visual elements over time on all layers independently indeed constitutes a new paradigm for creating moving images. And this is what After Effects and other animation, visual effects, and compositing software make possible today.[2] And while the paradigm of working with a number of layers placed on top of each other is itself not new – consider traditional cell animation, optical printing, photocollage, and graphic design – going from a few non-transparent layers to hundreds and even thousands, each with its own controls, fundamentally changes not only how a moving image looks but also what it can say.

But, innovative as it was, by the beginning of the 2000s the 2D digital compositing paradigm had already come to be supplemented by a new one: 3D compositing. The new paradigm has even less connections to previous media than 2D compositing. Instead, it takes the relatively new medium that was born with computers in the 1960s – 3D computer graphics – and transforms it into a general platform for moving media design.

The language used in professional production milieu today reflects an implicit understanding that 3D graphics is a new medium unique to a computer. When people use the terms "computer visuals", "computer imagery", or "CGI", which is an abbreviation for "computer generated imagery", everybody understands that they refer to 3D graphics as opposed to any other image source such as "digital photography". But what is my own reason for thinking of 3D computer graphics as a new medium – as opposed to considering it as an extension of architectural drafting, projection geometry, or set making? Because it offers a new method for representing physical reality – both what actually exists and what is imagined. This method is fundamentally different from what has been offered by the main media of the industrial era: still photography, film recording, and audio recording. With 3D computer graphics, we can represent the threedimensional structure of the world – versus capturing only a perspectival image of the world, as in lens-based recording. We can also manipulate our representation using various tools with ease and precision, which is qualitatively different from the much more limited "manipulability" of a model made from any physical material (although nanotechnology promises to change this in the future). And, as the case of contemporary architecture makes clear, 3D computer graphics is not simply a faster way of working with geometric representations such as plans and cross-sections used by draftsmen for centuries. When the generations of young architects and architectural students started to work systematically with 3D software such as Alias in the mid-1990s, the ability to directly manipulate a 3D shape (rather than dealing only with its projections as in traditional drafting) quickly led to a whole new language of complex non-rectangular shapes. In other words, designers working with the medium of 3D computer graphics started to imagine different things.

To come back to our topic of discussion: When the Velvet Revolution of the 1990s made possible to easily combine multiple media sources in a single moving-image sequence via digital compositing, CGI was added to the mix. Today, 3D models are routinely used in media compositions created in After Effects and similar software, along with all other media sources. But in order to be part of the mix, they need to be placed in their own 2D layers and thus treated as 2D images. This was the original After Effects paradigm: all image media can meet as long as they are reduced to 2D.[3]

In contrast, in 3D compositing paradigm all media types are placed within a single 3D space. This works as follows. A designer positions all image sources that are two inherently two-dimensional – for instance, digital film or digitized film, hand-drawn elements, typography – on separate 2D planes. These planes are situated within the single virtual 3D space. One advantage of this representation is that since 3D space is "native" to 3D computer graphics, 3D models can stay as they are, i.e. three-dimensional. An additional advantage is that the designer can now use all the techniques of virtual cinematography as developed in 3D computer animation. She/he can define different kinds of lights, fly the virtual camera around and through the image planes in any trajectory, and use depth of field and motion blur effects.[4]

3D compositing and the logic of reversal In 1996 I published the article "What is digital cinema?", which was my first attempt to describe the changes in the logic of moving image production I was witnessing (Manovich, 1996). In that article I proposed that the logic of hand-drawn animation, which throughout the twentieth-century was marginal in relation to cinema, became dominant in the computer era. Because software allows the designer to manually

manipulate any image regardless of its source, as though it was drawn in the first place, the ontological differences between different image media become irrelevant. Both conceptually and practically, they are all reduced to hand-drawn animation.

Having discussed the use of layers in 2D compositing using the example of After Effects, I can now add that animation logic also moves from the marginal to the dominant position in another way. The paradigm of a composition as a stack of separate visual elements as practiced in cell animation becomes the default way of working with all images in a software environment – regardless of their origin and final output media. In short, a moving image in general is now understood as a composite of layers of imagery. A "single layer image" such as un-manipulated digital video becomes an exception.

The emergence of the 3D compositing paradigm can be also seen as following the logic of temporal reversal. The new representational structure as developed within the computer graphics field – a 3D virtual space containing 3D models – has gradually moved from a marginal to the dominant role. In the 1970s and 1980s computer graphics were used only occasionally in a dozen feature films such as *Alien* (1979), *Tron* (1981), *The Last Starfighter* (1984), and *Abyss* (1989), and selected television commercials and broadcast graphics. But by the beginning of the 2000s, the representation structure of computer graphics, i.e. a 3D virtual space, came to function as an umbrella that can hold all other image types regardless of their origin. An example of an application that implements this paradigm is Flame, enthusiastically described by one user as "a full 3D compositing environment into which you can bring 3D models, create true 3D text and 3D particles, and distort layers in 3D space" (Okey, 2005).

This does not mean that 3D animation itself became visually dominant in moving image culture, or that the 3D structure of the space within which media compositions are now routinely constructed is necessary made visible (usually it is not). Rather, the way 3D computer animation organizes visual data – as objects positioned in a Cartesian space – became the way to work with all moving image media. As already stated above, a designer positions all the elements that go into a composition – 2D animated sequences, 3D objects, particle systems, video and digitized film sequences, still images and photographs – inside the shared 3D virtual space. There these elements can be further animated, transformed, blurred, filtered, etc. So while all moving image media have been reduced to the status of hand-drawn animation in terms of their manipulability, we can also state that all media have become layers in 3D space. In short, the new media of 3D computer animation has "eaten up" the dominant media of the industrial age – lens-based photo, film, and video recording.

FROM A "MOVING IMAGE" TO A "MEDIA COMPOSITION"

This is a good moment to pause and reflect on the very term of our discussion – moving image. When cinema in its modern form was born at the end of the nineteenth-century, the new medium was understood as an extension of an already familiar one – that is, as a photographic image that is now moving. This understanding can be found in the press accounts of the day and also in at least one of the official names given to the new medium – "moving pictures". On the material level, a film indeed consisted of separate photographic frames that when driven through a projector created the effect of motion for the viewer. So the concept used to understand it indeed fits with the material structure of the medium.

But is this concept still appropriate today? When we record video and play it, we are still dealing with the same structure: a sequence of frames. But for professional media designers, the terms have changed. The importance of these changes is not just academic and purely theoretical. Because designers understand their media differently, they are creating media that look different and have a new logic.

Consider the conceptual changes, or new paradigms – which at the same time are new ways of designing – we have discussed so far. Theoretically they are not necessary all compatible with each other but in production practice these different paradigms are used together. A "moving image" became a hybrid that can combine all different visual media invented so far – rather than holding only one kind of data such as camera recording, hand drawing, etc. Rather than being understood as a singular flat plane – the result of light focused by the lens and captured by the recording surface – it is now understood as a stack of separate layers potentially infinite in number. And rather than "time-based", it becomes "composition-based", or "object oriented". That is, instead of being treated as a sequence of frames arranged in time, a "moving image" is now thought of as a twodimensional composition that consists of a number of objects that can be manipulated independently. And finally, in yet another paradigm of 3D compositing, the designer is working in a three-dimensional space that holds both CGI and lens-recorded flat image sources.

Of course, frame-based representation did not disappear – but it became simply a recoding and output format rather than the space where the actual design is taking place. And while the term "moving image" can be still used as an appropriate description for how the output of a design process is experienced

by the viewers, it is no longer captures how the designers think about what they create. They are thinking today very differently than 20 years ago.

If we focus on what the different paradigms summarized above have in common, we can say that filmmakers, editors, special effects artists, animators, and motion graphics designers are working on *a composition in 2D or a 3D space that consists of a number of separate objects*. The spatial dimension became as important as the temporal dimension. From the concept of a "moving image" understood as a sequence of static photographs we have moved to a new concept: *a modular media composition*.

MOTION GRAPHICS

Let me invoke the figure of the inversion from marginal to mainstream in order to introduce yet one more paradigmatic shift. Another media type that until the 1990s was even more marginal to live action filmmaking than animation – typography – has now become an equal player along with lens-based images and all other types of media. The term "motion graphics" has been used at least since 1960 when a pioneer of computer filmmaking, John Whitney, named his new company Motion Graphics. However until the Velvet Revolution only a handful of people and companies had systematically explored the art of animated typography: Norman McLaren, Saul Blass, Pablo Ferro, R. Greenberg, and a few others.[5] But in the mid-1990s moving image sequences or short films dominated by moving animated type and abstract graphical elements rather than by live action started to be produced in large numbers. The material cause for the motion graphics take-off ? After Effects running on PCs and other software running on relatively inexpensive graphics workstations became affordable to smaller design, visual effects, post-production houses, and soon individual designers. Almost overnight, the term "motion graphics" became well known. The 500-year-old Gutenberg universe came into motion.

Along with typography, the whole language of twentieth-century graphical design was "imported" into moving image design. This development did not receive a name of its own, but it is obviously at least as important. Today (2006) the term "motion graphics" is often used to refer to all moving image sequences that are dominated by typography and/or design and embedded in larger forms. But we should recall that while in the twentiethcentury typography was indeed often used in combination with other design elements, for 500 years it formed its own word. Therefore I think it is important to consider the two kinds of "import" operations that took place during the Velvet Revolution – typography and twentieth-century graphic design – as two distinct historical developments.

DEEP REMIXABILITY

Although the discussions in this and the first part of this series of articles did not cover all the changes that took place during the Velvet Revolution, the magnitude of the transformations should by now be clear. While we can name many social factors that all could have and probably did play some role – the rise of branding, the experience economy, youth markets, and the Web as a global communication platform during the 1990s – I believe that these factors alone cannot account for the specific design and visual logics that we see today in media culture. Similarly, they cannot be explained by simply saying that the contemporary consumption society requires constant innovation, constant novel aesthetics, and effects. This may be true – but why do we see these particular visual languages as opposed to others, and what is the logic that drives their evolution? I believe that to understand this properly, we need to look carefully at media creation, editing, and design software and their use in production environment (which can range from a single laptop to a number of production companies collaborating on the same large-scale project).

The makers of software used in production usually do not set out to create a revolution. On the contrary, software is created to fit into already existing production procedures, job roles, and familiar tasks. But software programs are like species within the common ecology – in this case, a shared computer environment. Once "released", they start interacting, mutating, and making hybrids. The Velvet Revolution can therefore be understood as the period of systematic hybridization between different software species originally designed to do work in different media. At the beginning of the 1990s, we had Illustrator for making vector-based drawings, Photoshop for editing of continuous tone images, Wavefront and Alias for 3D modeling and animation, After Effects for 2D animation, and so on. By the end of the 1990s, a designer could combine operations and representational formats such as a bitmapped still image, an image sequence, a vector drawing, a 3D model, and digital video specific to these programs within the same design – regardless of its destination media. I believe that the hybrid visual language that we see today across "moving image" culture and media design in general is largely the outcome of this new production environment. While this language supports seemingly numerous variations as manifested in the particular media designs, its general logic can be summed up in one phrase: remixability of previously separate media languages.

As I have stressed in this text, the result of this hybridization is not simply a mechanical sum of the previously existing parts but new species. This applies both to the visual language of particular designs, and to the operations themselves. When

an old operation is integrated into the overall digital production environment, it often comes to function in a new way. I would like to conclude by analyzing in detail how this process works in the case of a particular operation – in order to emphasize once again that media remixability is not simply about adding the content of different media, or adding together their techniques and languages. And since remix in contemporary culture is commonly understood as these kinds of additions, we may want to use a different term to talk about the kinds of transformations the example below illustrates. Let us call it *deep remixability*.

What does it mean when we see depth of field effect in motion graphics, films, and television programs that use neither live action footage nor photorealistic 3D graphics but have a more stylized look? Originally an artifact of lens-based recording, depth of field was simulated in a computer when the main goal of the 3D computer graphics field was to create maximum "photorealism", i.e. synthetic scenes not distinguishable from live action cinematography.[6] But once this technique became available, media designers gradually realized that it can be used regardless of how realistic or abstract the visual style is – as long as there is a suggestion of a 3D space. Typography moving in perspective through an empty space; drawn 2D characters positioned on different layers in a 3D space; a field of animated particles – any composition can be put through the simulated depth of field.

The fact that this effect is simulated and removed from its original physical media means that a designer can manipulate it a variety of ways. The parameters that define what part of the space is in focus can be independently animated, i.e. set to change over time – because they are simply the numbers controlling the algorithm and not something built into the optics of a physical lens. So while simulated depth of field can be said to maintain the memory of the particular physical media (lens-based photo and film recording) from which it came, it became an essentially new technique that functions as a "character" in its own right. It has the fluidity and versatility not available previously. Its connection to the physical world is ambiguous at best. On the one hand, it only makes sense to use depth of field if you are constructing a 3D space even if it is defined in a minimal way by using only a few or even a single depth cue such as lines converging towards the vanishing point or foreshortening. On the other hand, the designer can be said to "draw" this effect in any way desirable. The axis controlling depth of field does not need to be perpendicular to the image plane; the area in focus can be anywhere in space and can also quickly move around the space, etc.

Following the Velvet Revolution, the aesthetic charge of many media designs is often derived from more "simple" remix operations – juxtaposing different media in what can be called "media montage". However, for me the essence of this Revolution is the more fundamental *deep remixability* illustrated by the example analyzed above. Computerization virtualized practically all mediacreating and -modification techniques, "extracting" them from their particular physical media and turning them into algorithms. (This means that, in most cases, we will no longer find any of these techniques in their pure original state.)

IMPORT/EXPORT: DESIGN WORKFLOW AND CONTEMPORARY AESTHETICS

In our discussions of the digital and After Effects interface and workflow (in part 1) as well as the newer paradigm of 3D compositing (this part), we have already come across the crucial aspect of the software-based media production process. Until the arrival of the software-based tools, to import media in different formats into a single space was either time consuming, or expensive, or in some cases simply impossible. As we saw, software tools such as After Effects have changed this situation in a fundamental way.

However, the contemporary software-based design of moving images – and any other media, for that matter – does not simply involve combining elements from different sources within a single application. In this section we shall look at the whole workflow typical of contemporary design – be it design of moving images, still illustrations, 3D objects and scenes, architecture, music, websites, or any other medium. (Of course most of the analysis of software-based design we have done so far in this and the previous article also applies to other media besides moving images. However, in this section I want to make this explicit, and therefore my examples below will include not only moving images.)

Although "import"/"export" commands appear in most modern media authoring and editing software running under GUI, at first sight they do not seem to be very important for understanding the software culture. You are not authoring new media or modifying media objects or accessing information across the globe, as in web browsing. All these commands allow you to do is to move data around between different applications. In other words, they make data created in one application compatible with other applications. And that does not look so glamorous.

Think again. What is the largest part of the economy of the greater Los Angeles area? It is not entertainment – from movie production to museums and everything in between (around 15%). It turns out that the largest part is the

import/export business (more than 60%). More generally, one commonly evoked characteristic of globalization is greater connectivity – places, systems, countries, organizations etc. becoming connected in more and more ways. And connectivity can only happen if you have certain level of compatibility: between business codes and procedures, between shipping technologies, between network protocols, and so on.

Let us take a closer look at import/export commands. As I will try to show below, these commands play a crucial role in software culture, and in particular in media design. Because my own experience is in visual media, my examples will come from this area but the processes I describe apply now to all media designed with software.

Before they adopted software tools in the 1990s, filmmakers, graphic designers, and animators used completely different technologies. Therefore, as much as they were influenced by each other or shared the same aesthetic sensibilities, they inevitably created differentlooking images. Filmmakers used camera and film technology designed to capture threedimensional physical reality. Graphic designers were working with offset printing and lithography. Animators were working with their own technologies: transparent cells and an animation stand with a stationary film camera capable of making exposures one frame at a time as the animator changed cells and/or moved background.

As a result, twentieth-century cinema, graphic design, and animation (I am talking here about standard animation techniques used by commercial studios) developed distinct artistic languages and vocabularies in terms of both form and content. For example, graphic designers worked with a two-dimensional space, film directors arranged compositions in threedimensional space, and cell animators worked with a "two-and-a-half" dimensional space. This holds for the overwhelming majority of works produced in each field, although of course exceptions do exist. For instance, Oscar Fishinger made one abstract film that involved moving three-dimensional shapes – but as far as I know, this is the only time in the whole history of abstract animation that we see an abstract three-dimensional space.

The differences in technology influenced what kind of content would appear in different media. Cinema showed "photorealistic" images of nature, the built environment, and human forms articulated by special lighting. Graphic designs feature typography, abstract graphic elements, monochrome backgrounds, and cutout photographs. And cartoons show hand-drawn flat characters and objects animated over hand-drawn but more detailed backgrounds. The exceptions are rare. For instance, while architectural spaces frequently appear in films because they could explore their three-dimensionality in staging scenes, they practically never appear in animated films in any detail – until animation studios start using 3D computer animation.

Why was it so difficult to cross boundaries? For instance, in theory one could imagine making an animated film in the following way: printing a series of slightly different graphic designs and then filming them as though they were a sequence of animated cells. Or a film where a designer simply made a series of hand drawings that used the exact vocabulary of graphic design and then filmed them one by one. And yet, to the best of my knowledge, such a film was never made. What we find instead is many abstract animated films that have certain connection to various styles of abstract painting. For example, Oscar Fishinger's films and paintings share certain forms.We can find abstract films and animated commercials and movie titles that have a certain connection to graphic design of the times. For instance, some moving image sequences made by motion graphics pioneer Pablo Ferro around the 1960s display psychedelic aesthetics that can be also found on posters, record covers, and other works of graphic design in the same period.

And yet, it is never exactly the same language. One reason is that projected film could not adequately show the subtle differences between typeface sizes, line widths, and grayscale tones crucial for modern graphic design. Therefore, when the artists were working on abstract art films or commercials that used design aesthetics (and most key abstract animators produced both), they could not simply expand the language of printed page into a time dimension. They had to invent essentially a parallel visual language that used bold contrasts, more easily readable forms and thick lines – which because of their thickness were in fact no longer lines but shapes.

Although the limitations in resolution and contrast of film and television image in contrast to printed page played a role in keeping the distance between the languages used by abstract filmmakers and graphic designers for most of the twentieth-century, ultimately I do not think it was the decisive factor. Today the resolution, contrast, and color reproduction between print, computer screens, and television screens are also substantially different – and yet we often see exactly the same visual strategies deployed across these different display media. If you want to be convinced, leaf through any book or a magazine on contemporary 2D design (i.e. graphic design for print, broadcast, and the Web). When you look at a spread featuring the works of a particular designer or a design studio, in most cases it is impossible to identify the origins of the images unless you read the

captions. Only then do you find that this image is a poster, that one is a still from a music video, and this one is a magazine editorial.

I am going to use Tashen's *Graphic design for the 21st century: 100 of the world's best graphic designers* (2001) to offer examples (Fiell & Fiell, 2003). Peter Anderson's images showing a heading against a cloud of hundreds of little letters in various orientations turns out to be the frames from the title sequence for a *Channel Four* documentary. His other image, which similarly plays on the contrast between jumping letters in a larger font against irregularly cut planes made from densely packed letters in much smaller fonts, turns out to be a spread from *IT Magazine*. Since the first design was made for broadcast while the second was made for print, we would expect that the first design would employ bolder forms – however, both designs use the same scale between big and small fonts, and feature texture fields composed from text that no longer need to be read. A few pages later we encounter a design by Philippe Apeloig that uses exactly the same technique and aesthetics as Anderson. In this case, tiny lines of text positioned at different angles form a 3D shape floating in space. On the next page another design by Apeloig also creates a field in perspective made from hundreds of identical abstract shapes.

These designs rely on software's ability (or on the designer being influenced by software use and following the same logic while creating the design manually) to treat text as any graphical primitive and to easily create compositions made from hundreds of similar or identical elements positioned according to some pattern. And since an algorithm can easily modify each element in the pattern, changing its position, size, color, etc., instead of the completely regular grids of modernism we see more complex structures that are made from many variations of the same element. Each designer included in this book was asked to provide a brief statement to accompany the portfolio of his/her work, and Lust has put this phrase as their motto: "Formfollows- process". So what is the nature of the design process in the software age and how does it influence the forms we see today around us?

Everybody who is practically involved in design and art today knows that contemporary designers use the same set of software tools to design everything. However, the crucial factor is not the tools themselves but the workflow process, enabled by "import" and "export" operations.

When a particular media project is being put together, the software used at the final stage depends on the type of output media and the nature of the project – for instance, After Effects for motion graphics projects and video compositing, Illustrator or Freehand for print illustrations, InDesign for graphic design, Flash for interactive interfaces and web animations, 3ds Max or Maya for 3D computer models and animations. But these programs are rarely used alone to create a media design from start to finish. Typically, a designer may create elements in one program, import them into another program, add elements created in yet another program, and so on. This happens regardless of whether the final product is an illustration for print, a website, or a motion graphics sequence; whether it is a still or a moving image, interactive or non-interactive, etc. Given this production workflow, we may expect that the same visual techniques and strategies will appear in all media designed with computers.

For instance, a designer can use Illustrator or Freehand to create a 2D curve (technically, a spline). This curve becomes a building block that can be used in any project. It can form a part of an illustration or a book design. It can be imported into an animation program where it can be set to motion, or imported into a 3D program where it can be extruded in 3D space to define a solid form.

Each of the type of programs used by media designers – 3D graphics, vector drawing, image editing, animation, compositing – excel at particular design operations, i.e. particular ways of creating a design element or modifying an already existing element. These operations can be compared to the different blocks of a Lego set. While you can make an infinite number of projects out of these blocks, most of the blocks will be utilized in every project, although they will have different functions and appear in different combinations. For example, a rectangular red block may become part of the tabletop, part of the head of a robot, etc.

Design workflow that uses multiple software programs works in a similar way, except in this case the building blocks are not just different kinds of visual elements one can create – vector patterns, 3D objects, particle systems, etc. – but also various ways of modifying these elements: blur, skew, vectorize, change transparency level, spherisize, extrude, etc. This difference is very important. If media creation and editing software did not include these and many other modification operations, we would have seen an altogether different visual language at work today. We would have seen "digital multimedia", i.e. designs that simply combine elements from different media. Instead, we see what I call "metamedia" – the remixing of working methods and techniques of different media within a single project.

Here are a few typical examples of this media remixability that can be seen in the majority of design projects done today around the world. Motion blur is applied to 3D computer graphics; computer-generated fields of particles are blended with live

action footage to give it an enhanced look; flat drawings are placed into a virtual spaces where a virtual camera moves around them; flat typography is animated as though it is made from a liquidlike material (the liquid simulation coming from computer animation software). Today a typical short film or a sequence may combine many such pairings within the same frame. The result is a hybrid, intricate, complex, and rich media language – or rather, numerous languages that share the basic logic of remixabilty.

As we can see, the production workflow specific to the software age has two major consequences: the hybridity of media language we see today across the contemporary design universe, and the use of similar techniques and strategies regardless of the output media and type of project. Like an object built from Lego blocks, a typical design today combines techniques coming from multiple media. More precisely, it combines the results of the operations specific to different software programs that were originally created to imitate work with different physical media (Illustrator was created to make illustrations, Photoshop to edit digitized photographs, After Effects to create 2D animation, etc.) While these techniques continue to be used in relation to their original media, most of them are now also used as part of the workflow on any design job.

The essential condition that enables this new design logic and the resulting aesthetics is *compatibility between files generated by different programs*. In other words, "import" and "export" commands of graphics, animation, video editing, compositing, and modeling software are historically more important than the individual operations these programs offer. The ability to combine raster and vector layers within the same image, to place 3D elements into a 2D composition and vice versa, and so on is what enables the production workflow with its reuse of the same techniques, effects, and iconography across different media.

The consequences of this compatibility between software and file formats, which was gradually achieved during the 1990s, are hard to overestimate. Besides the hybridity of modern visual aesthetics and the reappearance of exactly the same design techniques across all output media, there are also other effects. For instance, the whole field of motion graphics as it exists today came into existence to a large extent because of the integration between vector-drawing software, specifically Illustrator, and animation/compositing software such as After Effects. A designer typically defines various composition elements in Illustrator and then imports them into After Effects where they are animated. This compatibility did not exist when the initial versions of different media authoring and editing software initially became available in the 1980s. It was gradually added in particular software releases. But when it was achieved around the middle of the 1990s, within a few years the whole language of contemporary graphical design was fully imported into the moving image area – both literally and metaphorically.

In summary, the compatibility between graphic design, illustration, animation, and visual effects software plays the key role in shaping visual and spatial forms in the software age. On the one hand, never before have we witnessed such a variety of forms as today. On the other hand, exactly the same techniques, compositions, and iconography can now appear in any media. And at the same time, any single design may combine multiple operations that previously only existed within distinct physical or computer media.

CONCLUSION

In this second part of the series devoted to the analysis of the new hybrid visual language of moving images that today dominates our visual culture we have continued looking at how this language is shaped by software used in production. We have looked at how today the design of moving image sequences takes place within three-dimensional space, with 3D compositing gradually replacing 2D compositing. We have also looked at the shift from the concept "moving image" to a new concept of "media composition". Finally, we considered how the typical production workflow in a contemporary digital studio – the ways in which a project moves from one software application to another – affects contemporary visual aesthetics not only of moving but also still images.

Thus, we have moved from the description of how the new visual aesthetics looks visually to the analysis of the architecture of the key software application used since the middle of the 1990s to produce hybrid moving images (After Effects) to the discussion of how different applications interact with each other throughout the production process. Such a "theoretical workflow" provides one initial model of how to analyze "cultural software" – the applications used to access the Internet, view media, create websites, for product design, architecture, special effects, and all other types of cultural products, etc. In other words, this is one model to do what in the Language of New Media I have called "software theory".

Paradoxically, while social scientists, philosophers, cultural critics, and media and new media theorists have by now documented seemingly every aspect of our IT civilization, in the process generating a number of new fields and subjects (cyber culture, Internet studies, new media theory, digital culture), the underlying engine that drives most of these subjects

– software has received little direct attention. For most academics, artists, and cultural professionals interested in IT and its cultural effects, software is still invisible. But if we limit critical discussions to the notions of "cyber", "digital", "new media", or "Internet" we will never be able to get to what is behind new representational and communication media and to understand what they really are and what they do. If we do not address software itself, we are in danger of always dealing only with effects rather than causes: the output that appears on a computer screen rather than the software programs and professional cultures that produce it.

As we have already seen in the case of moving images, looking at software involved in their production goes a long way towards explaining why they now look the way they do. I strongly believe that without such analysis we will never be able to move beyond the commonplace generalities about contemporary colure – postmodern, global, remix, etc. – to actually describe the particular languages of different design areas and to understand the causes behind them. Thus, "software theory", which this series of articles practices, is not a luxury but a necessity. It is a new paradigm to decode culture that supplements other paradigms already in use.

NOTES

1. Soon after the initial release of After Effects in January 1993, the company that produced it was purchased by Adobe, which was already selling Photoshop.

2. Photoshop and After Effects were designed originally by different people at different time, and even after both were purchased by Adobe (it released Photoshop in 1989 and After Effects in 1993), it took Adobe a number of years to build close links between After Effects and Photoshop, eventually making it easy to go back and forth between the two programs.

3. I say "original" because in the later version of After Effects Adobe added the ability to work with 3D layers.

4. If 2D compositing can be understood as an extension of twentieth-century cell animation where a composition consists of a stack of flat drawings, the conceptual source of the 3D compositing paradigm is different. It comes out from the work on integrating live action footage and CGI in the 1980s done in the context of featurefilm production. Both film director and computer animator work in a three-dimensional space: the physical space of the set in the first case, the virtual space as defined by 3D modeling software in the second case. Therefore conceptually it makes sense to use three-dimensional space as a common platform for the integration of these two worlds. It is not accidental that NUKE, one of the leading programs for 3D compositing today, was developed in house at Digital Domain, which was co-founded in 1993 by James Cameron – the Hollywood director who systematically advanced the integration of CGI and live action in his films such as Abyss (1989), Terminator 2 (1991), and Titanic (1997).

5. For a rare discussion of motion graphics prehistory as well as an equally rare attempt to analyze the field by using a set of concepts rather than the usual coffee-table portfolio of individual designers, see Bellantfoni & Woolman, Type in motion.

6. For more on this process, see the chapter "Synthetic realism and its discontents" in Manovich, The language of new media.

REFERENCES

Bellantfoni, J., & Woolman, M. (1999). *Type in Motion*. New York: Rizzoli International Publications.

Fiell, C., & Fiell, P. (Eds.) (2001). *Graphic Design for the 21st Century: 100 of the World's Best Graphic Designers*. Cologne: Taschen.

Manovich, L. (1996, January 1). What is digital cinema? *Telepolis – das Magazin der Netzkultur*. Retrieved November 30, 2006, from http://www.heise.de/tp/r4/artikel/6/6110/2.html.

Manovich, L. (2001). *The language of new media*. Cambridge, MA: MIT Press.

Okey, A., (2005, December 28). Message posted to forums. *creativecow.net*. Retrieved October 10, 2006, from http://forums.creativecow.net/cgi-bin/dev_read_post.cgi?forumid/154&postid/855029.

CORRESPONDENCE

Dr Lev Manovich,
Professor, Visual Arts Department,
University of California, San Diego,
9500 Gilman Drive. #0084,
La Jolla, CA 92093-0084, USA.
E-mail: manovich@ucsd.edu

ISSN 1749-3463 print/ ISSN 1749-3471
DOI: 10.1080/17493460701206751
© 2007 Artifact