IMAGE
ZEITSCHRIFT FÜR INTERDISZIPLINÄRE BILDWISSENSCHAFT

Ausgabe 5 vom 1. Februar 2007

Themenheft

Computational Visualistics and Picture Morphology.

edited by Jörg R.J. Schirra

[Table of Content]

Jörg R. J. Schirra

Computational Visualistics and Picture Morphology – An Introduction

Abstract

Pictures have to be formalized digitally in an adequate manner when computer scientists are to work with them. It is mainly the relevant physical properties of the corresponding picture vehicle that have to be considered in that formalization: that is, the picture syntax. The present special issue of IMAGE deals in particular with morphological questions taking the specific, formalizing perspective of computational visualistics. It is also intended as the attempt to offer a clear and easily understandable summary of the state of the art of research on picture morphology in computational visualistics for picture scientists of the other disciplines. As an introduction, the relations between computer science, general visualistics, syntax studies, and morphology are examined.

Together with language, pictures have been connected to human culture from the very beginning (cf. [Schirra & Sachs-Hombach 2006a]). In the western societies they have gained a rather prominent place. However, steps toward a general science of images, which we may call ‘general visualistics’ in analogy to general linguistics, have only been taken recently (cf. [Sachs-Hombach & Schirra 2002], and [Schirra & Sachs-Hombach 2006b]). In computer science, too, considering pictures evolved originally along several more or less independent questions, which lead to proper sub-disciplines: *computer graphics* is certainly the most “visible” among them, but there are *image processing*, *information visualization*, and *computer vision*, as well. Only just recently, the effort has been increased to finally form a unique and partially autonomous branch of computer science specifically dedicated to images in general. In analogy to computational linguistics, the artificial expression ‘computational visualistics’ is used for addressing the whole range of investigating scientifically pictures “in” the computer (cf. [Schirra 2005]).

Pictures have to be formalized digitally in an adequate manner when computer scientists are to work with them. It is mainly the relevant physical properties of the corresponding picture vehicle that have to be considered in that formalization: that is, the picture syntax. The present special issue of IMAGE deals with exactly that theme taking the specific, formalizing perspective of computational visualistics. It is also intended as the attempt to offer a clear and easily understandable summary of the state of the art of research on picture morphology in computational visualistics for picture scientists of the other disciplines.

1 Computational Visualistics

Computational visualistics gains its name from its two parent disciplines: “computational” refers to the rather young discipline of computer science. “Visualistics” brings into mind the even younger unified science of pictures: general visualistics. Computer science, the endeavor of studying scientifically computers and information processing, has two different roots determining its methodology. In some aspects, computer science is a typical *structural science* like mathematics and logic: their subjects are purely abstract entities together with the relations in between. Such entities far off of our living practice are at best linked to everyday life by means of an interpretation relation arbitrary to the structures as such. With respect to some other aspects, computer scientists are like electrical engineers interested in *engineering* problems, an interest resulting in concrete artifacts that have already changed our lives dramatically during the past few decades and continue to do so with growing acceleration.

Correspondingly, the topics of computer science are, on the one hand, certain forms of purely abstract structures underlying data processing,[1] and on the other hand, certain kinds of purpose-bound artifacts we usually call “computers”. The concept “implementation” relates those two poles.[2]

---

[1] The processing of data is certainly a crucial theme for computer scientists, but it depends completely on the fact that data is always structured and grouped into types. Each such type implies a set of possibilities to “do something” with that kind of data: numbers can be added or multiplied (etc.); polygons in a geometric model can be moved or turned, mirrored or strained (etc.), but not *vice versa*. Usually, several data types and their interactions are relevant. As it is only important here that we can perform some operations with one sort of data so that certain relations hold between their results while ignoring the concrete manner of how those operations are actually realized, computer scientists consider *abstract data structures* – abstract entities that grasp exactly the essential properties. Algebraic formulae or logical expressions are often used to that purpose: the former for describing which operations transform the instances of which data type of the structure into what other type’s instances; and the latter determining which properties remain unchanged – invariant – after a certain sequence of operations; cf., e.g., [Ehrig & Mahr 1985].

[2] An implementation of an abstract data type – which is determined by a specification (a description) of its essential features – is a combination of more elementary data types, which are assumed unproblematic, so that the specification of the data type implemented is satisfied. If the data types employed for implementing are realized physically (e.g., in electro-technical devices), the implementation schema acts also as a plan for constructing a physical realization of the implemented data type. With such a physical realization, corresponding algorithms can be used to concretely manipulate instances of the data type. The implementation relation resembles the particular argumentation form expressed in synthetic judgments a priori by Kant (Critique of Pure Reason): In the mere specification, the essential features of an abstract data type remain contingent – like the axioms of a calculus. The implementation enables us to *found* those features: They are like

1.1 The Relation between Computer Science and General Visualistics

Quite obviously, *pictures* are not mentioned so far as a genuine topic of computer science. So, how are they linked with abstract data structures and their implementations on computers? That question is indeed a particular version of the more general problem of the relation between computer science and any domain of application; a relation that can be explained by means of the philosophical theory of rational argumentation (cf., e.g., [Ros 1999]) because the function of abstract data structures is equivalent to the function of the concepts structuring the rational argumentations in the domain of application. Data structures determine how formal expressions can correctly be constructed and transformed. The interrelated concepts that form a whole field of concepts[3] – computer scientists sometimes use the expression 'ontology' in this context, as well – determine how we ought to speak in a rational manner about a certain thematic domain, for instance about pictures, and how we may draw correct conclusions from corresponding assertions (in general visualistics, in our example).

The relation between computer science and any domain of application employs that equivalence. Applications of computer science to a certain subject are mediated essentially by means of a *formal translation* of the field of concepts that structures the rational argumentations in the application domain under investigation into a corresponding abstract data structure. Computational visualistics can thus be characterized by means of its central topic: the data structure(s) »image« that can be conceived of as the formalized equivalent(s) of the field(s) of concepts that form(s) the subject of general visualistics; or in other words: the former ruling formal expressions that are correctly constructed and transformed if and only if they correspond to the latter, which determine how we ought to speak in a rational manner about pictures. Algorithms in those data structures exemplify potential argumentations in picture theory in a formalized manner. Therefore computational visualistics is indeed able to contribute, as well, to general visualistics in return: with its algorithms implemented, the results of applying a theoretically proposed argumentation in a formalized and automatized manner onto concrete examples can be demonstrated and examined in great number with dramatically reduced effort. This is particularly evident in a range of picture phenomena that would even not exist without the help of computers: the interactive images.

---

that, *because* they are implemented in a specific manner on those data types with their particular features; cf. [Schirra 2005, Sections 2.1 & 4.3.1.2].

[3] If we refer by the expression 'the concept »*X*«' – e.g., by 'the concept »image«' – to everything that is structurally common to all explanations of '*X*' (in the example: the expression 'image') and its synonyms [Wittgenstein 1953] – that is, everything that "remains the same independent of how or in what language I formulate or show it" – then naturally, we never examine one concept alone: it is always a system of concepts that are mutually related and cannot be defined independently from each other, like »king«, »queen«, »knight, and »medieval society» (or alternatively »chess«) or, of course, »image« and »perception«. They belong to the same *field of concepts*.

1.2 Components of Computational Visualistics

Most of the pre-existing picture-related subjects in computer science focus on only certain aspects of the data structure »image«. In the area called *image processing*, the focus of attention is formed by the operations that take (at least) one picture (and potentially several other parameters that are not images) and relate it to another picture. With these operations, we can define algorithms for improving the quality of images (e.g., contrast reinforcement), and procedures for extracting certain parts of an image (e.g., edge finding) or for stamping out pictorial patterns following a particular Gestalt criterion (e.g., blue screen technique). Compression algorithms for the efficient storing or transmitting of pictorial data also belong into this field.

Two disciplines share the operations transforming images into non-pictorial data types. The field of *pattern recognition* is actually not restricted to pictures, but it has performed important precursory work for computational visualistics since the early 1950's in those areas that essentially classify information in given images: the identification of simple geometric Gestalts (e.g., "circular region"), the classification of letters (recognition of handwriting), the "seeing" of spatial objects in the images or even the association of stylistic attributes of the representation. That is, the images are to be associated with a non-pictorial data type forming a kind of description. The neighboring subject of *computer vision* is the part of AI (Artificial Intelligence) in which computer scientists try to teach – loosely speaking – computers the ability of visual perception. Therefore, a problem rather belongs to computer vision to the degree to which its goal is "semantic", i.e., the result approximates the human seeing of objects and their behavior in a picture.

The investigation of possibilities gained by the operations that result in instances of the data type »image« but take as starting point instances of non-pictorial data types is performed in particular in *computer graphics* and *information visualization*. The former deals with images in the closer sense, i.e., those pictures showing spatial configurations of objects (in the colloquial meaning of 'object') in a more or less naturalistic representation like, e.g., in a computer game. The starting point of the picture-generating algorithms in computer graphics is usually a data type that allows us to describe the geometry in three dimensions and the lighting of the scene to be depicted together with the important optical properties of the surfaces considered. Information visualizers are interested in presenting pictorially any other data type, in particular those that consist of non-visual components in a "space" of states: in order to do so, a convention of visual presentation has firstly to be determined – e.g., a code of colors or certain icons.

1.3 The Concept »Image«

The central issue of computational visualistics depends, in conclusion, on the core topic of general visualistics, i.e., the concept »image«. Correspondingly, determinations of that concept in image science are highly relevant for structuring the investigation of the data structure »image«, its algorithms, and the implementations thereof. It may therefore be rather helpful to end this section about computational visualistics with a short note on the concept »picture« in general visualistics.

Unfortunately, picture science has not yet come to final conclusions concerning the complete "ontology"[4] of pictures, which might be taken as the ultimate reference point for computational visualistics. Nevertheless, a sufficiently comprehensive determination to guide computer scientists dealing with pictures is available with Sachs-Hombach's [2003] proposition of a general conceptual framework, namely to determine the concept »picture« as »perceptoid signs«.[5] In the form of an Aristotelian definition with *genus proximum* (»sign«) and *differentia specifica* (»perceptoid«), this determination refers not only to two core aspects of pictures but opens originally, as we shall see below, the way to speak about pictorial syntax and picture morphology.

The superimposed concept »sign« implies that something – the *picture vehicle* – can be a picture if and only if it is in a certain way part of a special kind of situation that is characterized by a particular action: the sign act. That context also includes acting subjects called "sender" and "receiver". The sign (e.g., a picture) is used by the sender as a means to direct the focus of attention of the receiver onto something that is usually not present in that situation.[6]

Furthermore, in order to function properly each picture has to apply our abilities of visual perception in a specific manner, which we call its »perceptoid« character. More precisely, in using – i.e., adequately using – pictures we do not only perceive visually the sign in its physical appearance, that is, the picture vehicle. We have also to invoke – at least to some degree – our abilities to visually perceive spatial objects and configurations that are closely related with what the picture is employed to symbolize (the picture content).

---

[4] The term "ontology" is used here as in the context of computer science, i.e. equivalent to "field of concepts".

[5] The original German expression is "wahrnehmungsnahe Zeichen", cf. [Sachs-Hombach 2003, Sec. I.3].

[6] More precisely: sender and receiver are to be conceived of as roles that can also be simultaneously embodied by a single person. Correspondingly, we are able to bring something absent back into our own mind (and hold it in the focus of our attention) by means of "presenting a picture of it to ourselves" – only we say then plainly that we "look at the picture".

2 Pictures and Syntactic Investigations

Taking pictures to be a kind of sign allows the visualists – and that is, the computational visualists, too – to apply semiotic distinctions in order to guide their investigations. Since a picture like any sign depends on being part of a sign act, the broadest range of investigations (enclosing and determining all other questions) is the one that examines any relations between the other acts of sender and receiver with the signing activity – i.e., the presentation of a picture by a sender to a receiver in a certain context. That is the field of *pragmatics*. Examinations considering only the relations holding between the picture vehicles and what they are used to symbolize for sender and receiver determine the field of *semantics*.

Syntax is the third semiotic range of questions; and it is also the most restricted one since it deals with the sign vehicles (or in our case: the picture vehicles) alone. More precisely, the classifications of and relations between sign vehicles with respect to their physical properties are examined. This also includes the question of the range of variability of sign vehicles that may be used as the *same* sign, but also potential compositions of sign vehicles to more complex sign vehicles.

2.1 Syntactical Density

Syntactical considerations belong to the repertoire of picture theories since Nelson Goodman's publications at the latest (cf. [Goodman 1976], and also [Sachs-Hombach & Rehkämper 1999]). Although Goodman does indeed consider more than syntax, it is an important syntactic characterization of pictures that has had the most influence in general visualistics, so far. Syntactically, pictures are, he proposes, *dense* – in contrast to verbal signs, which are *syntactically distinct*. A sign system is called syntactically dense if the dimension of values for at least one of the syntactically relevant properties of the sign vehicles corresponds to the rational numbers: between any two values there are always more values. Sign vehicles with different values in that property are taken as different signs in that sign system. So, two of the infinitely many signs of such a system can be "infinitely similar" to each other, as there are always more sign vehicles "in between".

Syntactic characteristics of pictures are obviously defined by the visual properties of a marked surface of the picture vehicle. There are at least two different relevant dimensions that are apparently dense: (*i*) the positions of a point of color or a border between colors, and (*ii*) the perceived color (in a broad meaning). Between two different positions of a point of color, there is always – at least in theory – a (multitude of) position(s) in between. And similarly, in the theories of color two different color values are always connected by means of a sequence of intermediate color values, even if the human eye may not be able to distinguish those without the help of an artificial instrument.

The syntactically characteristic property of density is of high significance for the possibility of encoding, presenting, storing, and transferring pictures by computer. Is it decidable whether two pictures are syntactically equal? Can we, with other words, determine by means of effective, finite algorithms whether the transmission of a picture vehicle through the Internet, for example, has been correct, or whether a stored image still corresponds exactly to the original? Goodman has denied that possibility, which means that computational visualistics has a problem if he is right. Any computer system would only be able to differentiate picture vehicles up to a certain degree of resolution (in location or color).

2.2 Resolution in Computational Visualistics: Pixels

Indeed, the combination of images and computers did originally cost the former a property conceived of as characteristic for pictures by the scientists of many disciplines involved: pictures had to become digital in order to join that liaison. Essentially, ‘being digital’ means that the resolution of pictures has a definite (and often quite small) value. In contrast, the common view holds that picture vehicles have to be (at least in principle) analogous, i.e., without any limitation of resolution.

The most simple and well-known type for making picture vehicles available for a digital computer are bitmaps – matrices of *pixels* as they are called (‘picture elements’). This data type allows us to define a pixel-value for any pair of coordinates taken from two finite sets of successive indices (i.e., natural numbers). The pixel values encode a visual property, like color or intensity. Bitmaps have therefore a finite and fixed locale resolution that depends on the size a pixel is given: bitmap pictures are ratcheted. The number of different bitmaps of a given matrix size is finite, while the number of different matrix sizes is infinite but enumerable.[7]

The presentation of pictures on a computer screen typically employs this data type in just one matrix size. Although only a finite number of different picture vehicles is discriminated in that manner, an underlying data structure »image« still can be designed in order to fit the criterion of syntactic density imposed by general visualistics: the dense structure of a picture has to be projected (potentially only in parts) onto the syntactically distinct pixel matrix with the option of zooming in and out. In contrast to the visual approximations shown on the screen, a picture encoded by an instance of a data structure incorporating such a zoomable projection function needs not having a finite level of resolution (at least in

---

[7] Thus, although bitmaps are a rather limited candidate for the data type »image«, they have at least the advantage that there is no problem to decide identity or difference between two instances effectively.

theory: recall for example the small program systems fashionable few years ago that were used to visually inspect certain fractal functions, e.g., the Mandelbrot set).

Resolution is only one aspect of computational pictorial syntax: It corresponds roughly to the level of linguistics dealing merely with the range of letters; the notorious pixel usually comes into the beholder's (or creator's) focus of attention only when the presentation quality of a picture is low. There are other parts of which a picture vehicle is viewed as composed of and which could be rearranged to form another picture vehicle: When discussing syntactic design elements M. Scholz (1999), for example, refers to Paul Klee's pedagogic sketch book (1925, republished 1997) as an overview. Klee proposes several kinds of points, spots, lines, and areas (including typical geometric Gestalts like circle or square).[8] We shall later come back to such entities from geometry. Sometimes, candidates for syntactic elements can also be defined based on the production process: each stroke of a pen, a brush or a graving tool may lead to an individually visible mark usable as a syntactic element.

Of course, confronted with the questions of pictorial syntax and its combination rules, the first impulse of computer scientists is usually: to think of formal grammars.

2.3 Picture Grammars

Every computer scientist knows by heart the abstract structures called formal grammars – also called Chomsky grammars or compositional grammars or transformation grammars – since those are the major instrument for defining and classifying linear structures like programming languages. They are actually a tool from linguistics and have been applied to verbal syntax with great success.

A compositional grammar provides (*i*) a finite set of grammatical categories like 'article', 'prepositional phrase' or 'sentence', (*ii*) a lexicon (i.e., a collection of basic signs (words) each associated to a grammatical category), and (*iii*) a finite set of composition (transformation) rules. Essentially, each rule associates a grammatical category with a sequence of such categories, like in the following examples:[9]

PP → Prep + NP NP → Art + Noun NP → Art + Adj + Noun

---

[8] The major distinction in each of those element groups is that of an active, passive or medial element, depending on the role the design element plays in composition and production.

[9] In the examples, the usual labels 'PP' for prepositional phrase, 'NP' for noun phrase, 'Art' for article, 'Prep' for preposition, and 'Adj' for adjective are used. The rules can be employed mechanically in two ways: first, a given combined sign – a sentence – can be analyzed: it belongs syntactically to the language determined by the grammar if the sequence of syntactic categories that are associated to the words forming the sentence can be projected backward by means of several transformation rules to a special syntactic category (usually called 'sentence' or simply 'S'); second, starting from 'S', a number of applicable combination rules is used in forward direction in order to synthesize a list of syntactic categories that can be associated to words in the lexicon generating a well-formed sentence.

Those three sets determine all sentences, i.e., sequences of words, belonging to the language considered. Note that each word listed in the lexicon always has clear semantic and grammatical functions of its own.

Assuming that all pictures form just one “language”,[10] a formal grammar for picture syntax thus would also have to provide corresponding sets of syntactic categories, elementary pictures with associated syntactic categories, and composition rules. Those set should be accordingly applicable for analyzing in a mechanical manner given objects in order to decide whether they are pictures,[11] or to generate from the starting category any picture vehicle. Such a formal grammar for pictures would indeed enable us to distinguish between well-formed and ill-formed picture vehicles.

Unfortunately, all proposals so far to provide such a combinational grammar system for all pictorial signs (or even large subsets) have failed: only very special pictorial media – that apparently are also used in a way similar to language anyway, like pictograms – could be formalized in that manner.[12] In general, there does not even seem to be anything like an ill-formed picture vehicle at all (cf. [Plümacher 1999]). Any more or less flat surface that can be visually perceived can apparently serve as a picture vehicle.

Already the question “what are the syntactical elements in the ‘lexicon’ – as we do not have a better expression, so far – of copper engravings (for example)” is not easily answered. Can the engraving lines carry that function? Are pixels – as used in computer visualistics – better candidates? However, neither engraving lines nor digital pixels bear a proper pictorial meaning by their own – one of the characteristics in the linguistic case, i.e., for the words in the lexicon.

Furthermore: What corresponds to the grammatical categories? Are perhaps “Circle” or “Spot” pictorial analogies of “Noun” and “Art”? And if so, what would actually be the difference between the ‘lexical’ basic elements and the grammatical categories in that system?

In conclusion: Being rather fertile in linguistic syntax studies, the idea of generative syntax has often been proposed for pictorial syntax, as well – though, with little success: compositional syntax is mainly interested in the syntactically correct composition of words (as elementary verbal signs) into sentences (i.e., compound verbal signs). A pictorial analogy of words so that pictures could be conceived of as corresponding sentences has not been suggested in a convincing manner. However, another important building block of syntax studies – at least in linguistics – is given by morphology.

---

[10] Alternatively, several pictorial subsystems may be syntactically distinguishable.

[11] – or belong to the particular pictorial subsystem in question.

[12] Similarly, arrangements of pictures as in journal layout or comics, and even the sequences of scenes in film can partially be analyzed by means of formal grammars.

3 Morphology

3.1 Morphology in Linguistics

In linguistic morphology, the rules of building words, and hence the inner structure of words is examined instead of sentences.[13] Words are partitioned in segments called 'morphemes'[14] that contribute to the word's meaning or grammatical function. The postfix '-ed' in English, the prefix 'pré-' in French, or the root '-wend-' in German are typical examples for morphemes. Mostly, morphological elements are identified and arranged into classes by means of a rule of mutual exchange: some words beginning with 'pré-' can be transformed into other words of French by just changing the prefix to 're-', 'con-', 'de-' etc.

More generally, morphological modifications can be differentiated into internal modifications mainly by means of vowel permutations (e.g., 'come' to 'came'), and external modifications by means of affixes – beside prefix and postfix, some languages also use infix and circumfix modifications. While inner modification alters the "color" of a word, so to speak, external modification changes its shape and size. Thus, the combination of morphological elements also plays a major role in the invention of completely new words.

Morphemes do not have to be – and are usually not – words by themselves.[15] Even the semantic or grammatical function of one morpheme can be ambiguous and may change in different compositions (e.g., "s" as flexion postfix and plural postfix in English). Morphemes may best be viewed as the vehicles of unsaturated partial signs acts without an independent pragmatic function[16] that modify in a more or less specific way the meaning of the whole.

There are arguments that syntax in the form of a formal grammar, and syntax as morphology are not categorically opposed but form the two ends of a more or less continuous scale of various language structures: from the *analytic* language structure (also: isolating languages) to the various types of *synthetic* language structures (with the subsets of agglutinating, flexing, and fusing languages), and finally the *polysynthetic* language structure (also: incorporating languages).[17] In an extremely isolating language (like Chinese), words are never ever modified. All grammatical relations are expressed by special words. Sen-

---

[13] Although the expression "morphology" was already introduced to linguistics by August Schleicher in 1859 (under the influence of Goethe's morphological theory of plant growth), a specific morphological investigation of words – in contrast to syntacto-grammatical studies of sentences and apart from phonology – did not become prominent before the 1970s.

[14] The term "morpheme" was proposed around 1881 by B. de Courtenay and elaborated by L. Bloomfield.

[15] Morphemes that are also words are called *free*; the other morphemes are *bound*. As free morphemes are listed in the lexicon, they are also called *lexemes*.

[16] This is in contrast to predication or nomination, which are also unsaturated partial sign acts but each carrying a quite specific pragmatic function (of introducing a distinction to the discourse universe, or naming a discourse object respectively).

[17] The distinction was already introduced by W. v. Humbold and A. W. v. Schlegel.

matgībulhahumš

<table>
<tr><td colspan="2"></td><td>*(gāb)*
gīb</td><td colspan="4"></td></tr>
<tr><td></td><td>*t(i)*</td><td>…</td><td>*u*</td><td colspan="3"></td></tr>
<tr><td colspan="4"></td><td>*l(ī)ha*</td><td colspan="2"></td></tr>
<tr><td colspan="5"></td><td>*hum*</td><td></td></tr>
<tr><td>*ma*</td><td colspan="5">. . .</td><td>*š*</td></tr>
</table>

Schema 1

tences are straightforward groupings of words usually in a relatively strict word order; correspondingly, a separate study of morphology does not make sense. An extremely polysynthetic language would in contrast consist of one-word sentences only, a single word that may consist of many morphemes all melted together in order to modify the complete meaning accordingly. Thus, syntactical investigations here are purely morphologic.[18]

The morphological structure of the word *matgībulhahumš* in Egyptian Arabic,[19] for example, could be literally translated to approximately "not-you-all-ought-bring-her-them-thing" (i.e., "do not bring them to her, all of you"). It consist of the two circumfixes *ma...š* ("not … thing") and *t(i)...u* (marker for 2nd person plural imperfect in jussive mode: approx.: "you ought to"), the two morphemes *l(ī)ha* (3rd person singular feminine dative), *hum* (3rd person plural accusative), and, as the root, an internally modified lexeme *gīb* (the imperfect form of *gāb*: "to bring"), as is indicated in schema 1.

All the morphological elements are fused to a single word that is used as a sentence. The schema of such complicated combinations by means of the fusion of morphemes with partial phoneme elisions – together with the used of enclosing or inserting affixes – can indeed much stronger evoke the idea of a syntactic structure of pictures than the schema of formal grammars.

3.2 Transfer to Visualistics

Intuitively, the system of pictures and most of its subsystems are similar to extremely polysynthetic languages. Of course, picture vehicles do have parts that modify the pictorial meaning and use of that vehicle. But for each picture, those parts are closely fused together – comparable to an enormously complex one-word sentence. They form a single

---

[18] Many Native American languages like Náhuatl are more or less strongly incorporating. Flexing and agglutinating languages are somewhere between the two extremes. The word order is usually not as restricted as in isolating languages, and a mixture of grammatical and morphological rules determines the syntactic structure.

[19] Linguists report that the Egyptian version of Arabic has a strong tendency to polysynthetic structures in contrast to high Arabic.

entity that does not allow us usually to isolate in a clear manner the semantic contribution of any part, as it had to be expected in the case of a formal grammar. Nevertheless it is clear that any morphological element of the picture vehicle – or *pixeme* for short – does contribute in some way to the meaning, and hence modifies the use of the picture. Therefore, any tiny change in the spatial distribution of pigments may very well be seen primarily as a modification in the sense of morphology.

There are several characteristic differences to verbal morphology: In contrast to the essentially temporal and hence linear composition of verbal morphology, pictorial morphology extends in (flat) space and thus in (usually) two coordinated dimensions, which increases the complexity quite heavily. Instead of the pair of possible directions for morphological extensions – "before" (as prefix) and after (as postfix)[20] – an infinite and actually dense multitude of directions can be used to position pixemes.

Of course, the specific difference of resolution already mentioned above has to be taken into account, as well: there is a distinct lower limit to resolution in linguistics since morphemes cannot be smaller than the difference between two letters or phonemes. For picture vehicles, no such quantization is evident. The criterion of density also implies that any pixeme can – at least in principle – be considered as composed of even smaller pixemes.

Brush strokes, pencil lines, etc. are rather good candidates for simple pixemes, as was already mentioned above.[21] They are composed into more complex configurations that nevertheless still are pixemes. In general, we may view any geometrical entity of two-dimensional geometry of the picture vehicle as a pixeme. Then, even a picture is a pixeme, as well – which makes sense as its surface can be seamlessly incorporated in another picture vehicle. Still, pictures may very well have a morphological structure without a list of given elements that are pictures themselves. Although there appears to be no (natural) verbal language that employs bound morphemes only, morphology does not necessarily depend on the existence of free morphemes (lexemes). On the other hand: there always exists a maximal pixeme to which all the other pixemes are infixes. It is the frame that externally binds and thus determines the maximal pixeme. Indeed, maximal pixemes might act as free morphemes for picture vehicles.[22] While verbal structures grow morphologically outward by adding elements mostly externally, pictorial structures grow morphologically inward by adding details internally.

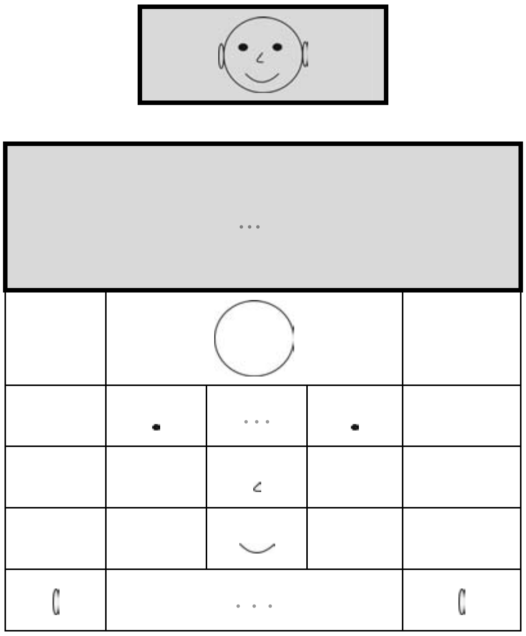
Since morphemes essentially change the color of vowels in the course of an internal modification, a literal change of color of a pixeme is a very plausible candidate for the corre-

---

[20] Circumfixes employ accordingly both directions, and infixes can be seen as inverse circumfixes.

[21] See also the contributions of Engelhardt and of Isenberg in this volume.

[22] As another hint for a kind of free pixemes the following psychological evidence may be counted: a schema corresponding to an elementary face pixeme (or rather a set of affect-expressing face pixemes) is inborn to all human beings and already effective for very young children.



Schema 2

sponding derivation. Again, the bandwidth of alternatives is characteristically different: a finite set of phonemes vs. the colors from a dense range of options.

Evidently, the rules of visual perception are constitutive for the “segmentation” of pictures in pixemes.[23] The empirical findings from psychophysics and the concepts of Gestalt theory in particular help to determine the laws of pixeme formation. The former indicate general principles of indiscernibility of optical properties while the latter formulates grouping principles that bind compound pixemes to the constituting simpler pixemes. That decomposition runs down to optically uniform regions, which we find on any level of resolution since we deal with dense fields both in color and in location. An optically uniform region is not only given by a single color, but also by a color gradient (in particular a saturation or intensity gradient), and even by homogenous textures.

As an extremely simplified example in analogy to the verbal example above, schema 2 exemplifies a morphological (de)composition for a picture. In accord with the assumption mentioned above that pictorial morphology grows inward, the frame defines the root in the decomposition, or more precisely: bound by the frame, the empty “canvas” acts as the maximal pixeme. As an infix, the face marks modify the maximal pixeme. The face mark itself consists of a simple circular pixeme with several infixes and one circumfix (the ear marks).

Of course, the specific difference of resolution already mentioned above has to be taken into account, as well: there is a distinct lower limit to resolution in linguistics since morphemes cannot be smaller than the difference between two letters or phonemes. For pic-

---

[23] See also the contributions of du Buf and Rodrigues, and of Hermes and the SVP Group in this volume.

ture vehicles, no such quantization is evident. The criterion of density also implies that any pixeme can – at least in principle – be considered as composed of even smaller pixemes.

Although there is still much more to be said about pictorial morphology in general, it is now the time to come back to the particular perspective of computational visualistics.

4 Aspects of Picture Morphology in Computational Visualistics

Morphological considerations in the particular context of computational visualistics are at the focus of this thematic part of IMAGE V. We are interested in questions like the following: What alternative formalizations for pixemes apart from pixels can be offered by computational visualistics? Where and in which form do such formal pixeme systems play an important role? And what is the influence these formalizations in computational visualistics have on picture morphology in general?

4.1 Some Specific Approaches

Let us concentrate for the moment on lines or strokes. A stroke may be defined pragmatically by the painter’s movement or semantically as the contour line of an object. Beside the potential graphical meaning of a line or the stylistic indications associated with its particular make (not to mention any other expressive or appellative function of dynamism associated to it on the level of pragmatics), there are several dimensions in which a line – just being taken as a line – can vary: most prominently in the course or path it takes. But there are other ranges: is it a continuous line, or dashed, or dotted? Does it consist of strokes of one kind or another? How thick is it? Does its thickness change over its course or not? Is there an internal fine structure to the strokes?

An extensive treatment of data types for strokes and lines and their possible implementations has been performed in the context of non-photorealistic rendering (NPR), a sub field of computer graphics.[24] While Figure 1 exemplifies several types of digital “hairy brush strokes” that have been generated – quite expensively in computational resources – by simulating a brush with several individual bristles applied with changing pressure to a certain kind of surface, Figure 2 shows examples of lines resulting the application of a “style function” to the “skeletal path” of the stroke.[25] Both constituents of the latter case are defined by means of parametric curves: the style describes how a given path (as the core of the line) is to be perturbed in order to result in a corresponding pixeme. Style and path can be viewed as independent ranges determined in each particular picture by semantic and / or pragmatic aspects.

---

[24] See also the contributions of Isenberg, and of du Buf and Rodrigues in this volume.

[25] Figure 1 was quoted from [Strassmann 1986], Figure 2 from [Schlechtweg & Raab 1997].

Figure 1: Enlarged Fine Structure of Computer-Generated Stroke Types

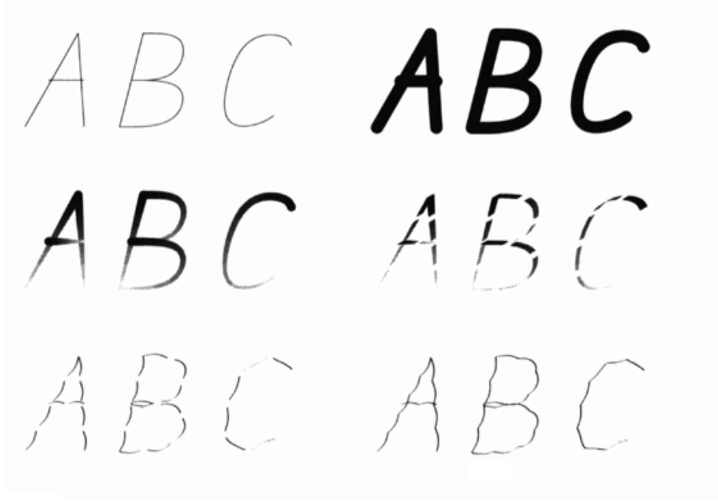


Figure 2: Examples with Style-Parameterized Stroke Functions

To some degree, the rules of composition of strokes or other pixemes into a picture can be investigated by means of the tools of formal languages. Formal grammars based on replacement rules that lead to two-dimensional “pictorial” structures have been investigated essentially under the name of L-systems. The expressions generated by an L-system can be interpreted as orders to place substructures, and to move or turn in-between. A fairly simple example is defined by the following replacement rule:

$$P \rightarrow P\,[-P\,]\,P\,[+P\,]\,P$$

Interpret “P” as “place a pixeme and move a bit forward”, “+” by “turn right”, “–” by “turn left”, and the square brackets as stack operations that allow us to return to that point after the bracketed sub expression has been dealt with. The plant-like structures in Figure 3 have been generated by this rule. Obviously the pixemes themselves are not really relevant for L-systems and their relatives, since these grammars basically deal with arrangements and groupings of abstract entities that may or may not be interpreted in a pictorial sense.[26]

For a more extensive approach to pictorial morphology, a data type for pixemes can best be derived from a calculus for geometry. That any pixeme must be a geometric entity seems almost too trivial to be mentioned. That inversely any entity in flat geometry – apart from non-extended points – may also be a candidate for a pixeme is at least a good guess. Taking the common Euclidean formalization of geometry leads however to the “unpleasant” consequence that the most basic pixemes must be non-extended points – a concept highly abstracted from experience, that is.[27]

---

[26] See also the contribution of Kurth in this volume.

[27] See also the contribution of Engelhardt in this volume.

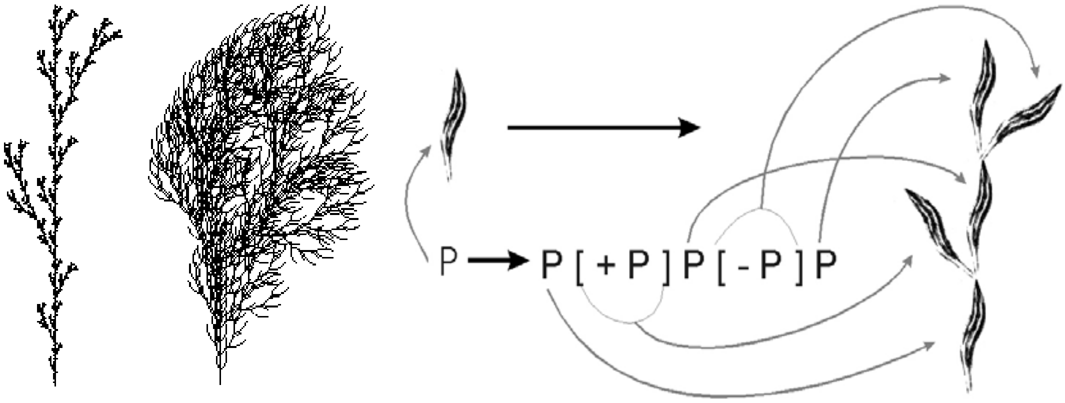


Figure 3: Two Complex Example Morphemes Generated by (Bracketed) L-Systems, and the Graphical Interpretation for the Rule for the Left Example

Fortunately, some non-standard approaches to geometry offer an interesting way out. The traditional calculus of geometry develops around the fundamental concept of a zero-dimensional point. In contrast, the family of *mereogeometries*[28] is based on extended regions as the most elementary entities, which may or may not have (distinguishable) proper parts. The regions are often called “individuals”. Individuals do not have immediate attributes of form or position: only the relations to other individuals, in particular parts, determine form and (relative) location.

An individual may quite well be thought of as a visual Gestalt – thus following the principle of perception psychology of the Gestalt school: one has to consider the perceived whole first and introduce the concepts for perceptual atoms as instruments of the explanations of the former, not the other way round. We do not see sets of zero-dimensional points but regional Gestalts. The abstract notion of a spatial entity without extension is secondarily constructed in order to explain some aspects of experienced space, but leads on the other side to severe difficulties as the discussion on infinite resolution has shown. Therefore, the constructs of an individual calculus for the two-dimensional mereogeometry are excellent candidates for a general and exhaustive discussion of pixemes.

In fact, the concept of a minimal region can be introduced in mereogeometry: They are usually called a “point”, but we may well use “pixel” instead. A point in this sense is a region that has no proper parts (or rather, a region where no proper parts are considered). When the concept »point« is introduced in the data structure in that manner, there is no need in any concrete instance for using infinitely many point instances: only the “relevant” points must be instantiated. This also means that there is always a finite resolution.

---

[28] See also the contribution of Borgo *et al.* in this volume.

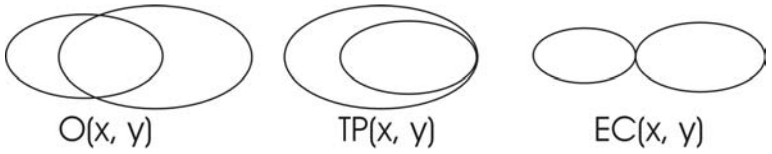
Mereogeometrical Calculi of Space

Mereogeometries are a group of particular logical formalisms for describing n-dimensional space that are based on mereological calculi. Mereology's focus of interest is part-whole relations. For instance, Clarke's mereological calculus (1981; here quoted from Vieu 1991, 120ff) is based on the primitive relation C(x, y) with the intended meaning "individual x is connected with individual y" and defined by the axioms:

A0.1	$\forall x \, (C(x, x) \wedge \forall x \, \forall y \, (C(x, y) \Rightarrow C(y, x)$	Reflexivity and symmetry
A0.2	$\forall x \, \forall y \, (\forall z \, (C(z, x) \Leftrightarrow C(z, y)) \Rightarrow x=y)$	Axiom of extension

Some definitions possible in Clarke's calculus are …

D0.1	DC(x, y)	$\equiv_{def}$	$\neg \, C(x, y)$:	"x is disconnected with y"
D0.2	P(x, y)	$\equiv_{def}$	$\forall z \, (C(z, x) \Rightarrow C(z, y))$	"x is part of y"
D0.3	PP(x, y)	$\equiv_{def}$	$P(x, y) \wedge \neg \, P(y, x)$	"x is a proper part of y"
D0.4	O(x, y)	$\equiv_{def}$	$\exists z \, (P(z, x) \wedge P(z, y)$	"x overlaps y"
D0.6	EC(x, y)	$\equiv_{def}$	$C(x, y) \wedge \neg \, O(x, y)$	"x is externally connected with y"
D0.7	TP(x, y)	$\equiv_{def}$	$P(x,y) \wedge \exists z \, (EC(z, x) \wedge EC(z, y))$	"x is a tangential part of y
D1.1	x=F(α)	$\equiv_{def}$	$\forall y \, (C(y, x) \Leftrightarrow \exists z \, (z \in \alpha \wedge C(y, z))$	"x is identical to the fusion of the set of individuals α"
D1.2	x+y	$\equiv_{def}$	$F(\{z : P(z,x) \vee P(z, y)\})$	"x+y is the sum of x and y"
D1.5	x∧y	$\equiv_{def}$	$F(\{z : P(z,x) \wedge P(z, y)\})$	"x∧y is the intersection of x and y'



… so that, for instance, the following theorem can be proven:

T0.34 $\forall x \, \forall y \, \forall z \, ((TP(z, x) \wedge P(z, y) \wedge P(y, x)) \Rightarrow TP(z, y)$

A definition of "point" out of a set α of individuals (with Λ being the empty set):

$$
\begin{aligned}
PT(\alpha) \equiv_{def} \; & \neg \, \alpha=\Lambda \;\; \wedge \;\; \forall x \, \forall y \, ((x\in\alpha \wedge y\in\alpha) \Rightarrow (EC(x, y) \vee (O(x, y) \wedge x\wedge y\in\alpha))) \;\; \wedge \\
& \forall x \, \forall y \, ((x\in\alpha \wedge P(x, y)) \Rightarrow y\in\alpha) \;\; \wedge \;\; \forall x \, \forall y \, (x+y\in\alpha \Rightarrow (x\in\alpha \vee y\in\alpha))
\end{aligned}
$$

(*i.e., all individuals partaking in a point are connected with each other; if two of them overlap, their intersection is also part of the point; each individual containing an element of the point is also element of that point; and finally, if an element of the point is the sum of two individuals*)

This calculus already allows dealing with topological relations and can be extended easily to a full geometry (i.e., including directions and metric distance). That is, any geometrical configuration can be described by a set of propositions of that calculus. Any analysis or transformation of the geometrical configuration can correspondingly be performed in analogy with the set of propositions by means of logical analyses or transformations (cf., e.g., Pratt-Hartmann 2000]).

While Euclidean geometry first introduces the continuous range of infinitely many coordinates determining potential points some of which are then chosen to be relevant (still an infinite number in any practical relevant instance), mereogeometry starts with a (usually finite) number of relevant individuals (regions) we can think of being given in perception. That is, we may indeed assume that the principles governing visual perception determine the regions that are syntactically relevant, hence leading only to the essential "points" determined by the given individuals.

The empty picture plain – as the simplest maximal pixeme – is particularly characterized in its most usual rectangular form by the four corner points. The "energetic field" often associated to such a maximal pixeme (cf. Fig. 4) cannot easily be derived as it depends essentially on features of the perceptual mechanism not covered by the Euclidean calculus as such.[29] Additional explanations have to be added that often employ rather mystical metaphors to physics.[30] The mereogeometrical conception of points and limits may offer a better access to the problem of the "energetic aspects" of pixemes, and especially of the empty picture plane: As those points are only conceivable as the result of operations on extended regions, the four corner points implicitly refer to defining individuals (virtual pixemes). It is a promising hypothesis for future research to derive within the calculus of mereogeometry any "energetic effects" from those implicit pixemes.

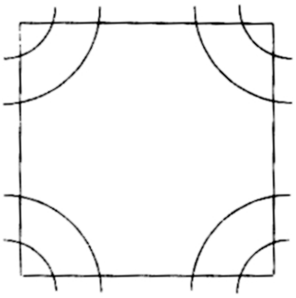


Figure 4: Rectangular maximal pixeme with "energetic phenomena" as sketched by Saint-Martin [1990, 97]

Mereogeometries are a formal way to deal with geometry in a manner more closely related to visual perception than traditional point geometry. If we accept the view that the central data type of a two-dimensional mereogeometry determines what is a pixeme – namely any connected sub system of individuals, then there is indeed no finite number of possible pixemes – a clear difference to verbal sign systems with their strictly limited number of morphemes. However, any pixeme can be described and dealt with in a unique and generatable manner in the calculus in a finite number of steps: pixemes can be combined to form pixemes of a higher order – until every visually separable Gestalt of a picture is covered.

---

[29] cf. Saint Martin 1990, 96: "By reason of its dynamic origin, this Basic Plane must be defined as an energy-charged portion of space, generated by the radiating energies produced by the angular intersections of the four straight lines. It is through this maximal energizing of right angles that a dynamic structure emerges and is propagated to form a Basic Plane. Irrespective of the physical characteristics of the material support which facilitate its deployment, the Basic Plane is defined as an ensemble of energetic phenomena, taking its point of origin in the peripheral lines and corners that envelop and contain it. Its energetic and topological characteristic will remain the essential element which determines the spatial structure of the Basic Plane".

[30] *ditto*, p. 97: "While essentially describable as the interplays of various levels of intensity of energy, perceptual systems are animated by the different categories of actual, potential, and virtual energies offering a decreasing order of forces. The actual and potential levels are established by the contribution of both the visual elements and perceptual processes, the virtual being the unique product of perceptual activity".

4.2 The contributions of the volume

Since the fluctuation of the focus of attention between structural science and engineering is characteristic for all investigations in computer science, it is also valid for the dealing with pictorial data. On the one hand, particular abstract data types for pictorial representations are investigated and designed from a purely structural point of view. For example, efficiency properties are examined, or minimal sub-structures for particular tasks determined. On the other hand, concrete algorithms for, e.g., picture processing are "software-engineered" and used in diagnosis. Correspondingly, the papers collected in this issue exhibit a wide range between analytic investigations and constructive engineering.

The call for paper for this thematic issue of IMAGE did in particular list the following five 'crystallization cores' for a discussion of picture morphology from the perspective of computational visualistics:

- Picture morphology as Grammar: L-Systems and Similar Formalizations
- Mereo-Geometrical Approaches to Picture Morphology
- Pixemes in Non-Photorealistic Computer Graphics
- Image Processing: Pixeme-based Approaches of Picture Manipulation and Computer Vision
- Glyphs and Icons: Pixemes in Information Visualization

With the exception of the fifth theme – each item has been covered by at least one contribution.

The first two texts deal with the general question of the systems of pictorial syntax or morphology and its constituents. A set of building blocks for formally describing graphics is presented in the contribution of **Engelhardt** (Netherlands). He takes a perspective rather related to design and design theory, and proposes a set of building blocks for all graphics derived from the relevant literature. Three types of building blocks are distinguished: graphic objects, meaningful graphic spaces, and graphic properties. Although this system does not yet reach the formal stringency of the logical calculi employed, for instance, in the formal ontology of space, it provides a good entry point for the discussion of computational picture morphology.

An overview on the formal representations of space in the field of formal ontology, a subdomain of AI and cognitive science, is given by the contribution of **Borgo** and colleagues (Italy). Without putting too much stress on the (actually rather demanding) underlying logical and mathematical formalizations, these authors explain the advantages of mereo-geometrical approaches in the cognitive dimension fitting the qualitative categorizations of the human access to space. From that perspective, they discuss the application of mereo-geometrical calculi to the description of pictorial morphology. While Engelhardt starts from more or less informal notions as used in design theories and proposes a systematic categorization of graphic objects, rules for their combination, and a typology of meaningful graphic space, the Italian group moves from highly formalized concept (which are elaborated in formal calculi) toward the more informal notions employed in pictorial syntax.

With his survey on morphological models with L-Systems and relational growth grammars, **Kurth** (Germany) brings into the debate another meaning of the expression 'morphological' – a meaning more closely related to the word's original, i.e., biological context: the knowledge about the bodily forms of living beings, and the rules of the arrangements of body parts and organs (especially in the temporal development). The special grammatical formalisms described by Kurth do not originally refer to pictures but to objects that are conceived of as being constructed by formally arranging parts in space by means of a quasi-biological manner of "growing", and that are often depicted in order to be further studied or used. Therefore, this meaning of 'morphological' actually exceeds the borders of strict syntax – after all, the structures of the things depicted are actually in the range of semantics. Nevertheless, the formal options given by means of quasi-grammatical mechanisms for "growing" spatial arrangements of "body parts", and the geometrical arrangement of pixemes are close enough to the discussion on pictorial morphology for further enlightening the latter.

While Kurth is more interested in the arrangement, i.e., the spatial configuration of any kind of parts, the contribution of **Isenberg** (Canada), turns our focus of interest to the potential parts to be arranged by giving an overview on the techniques used to generate computer graphics apart from naturalistic – say: "photo-realistic" – representation styles. Contrasting the resulting images with the photorealistic case, Isenberg describes a wide range of morphological modifications possible with those techniques. Different rules for calculating shading, for example, lead to a picture that is internally modified; applying strokes or graftals corresponds to external modifications. Unlike the pixel, the morphological primitives used in NPR often carry a "meaning" beyond the syntactical level; saving the morphological structure with the picture is therefore, so Isenberg, often quite helpful for subsequent processing.

The paper of **du Buf** and **Rodrigues** (Portugal) also aims for non-photorealistic rendering, as the authors explain how computational models of neuro-physiological explanations of visual perception can be employed in order to generate painterly pictures. After giving us an overview about the relevant state of the art of neuro-physiological analyses, they consider the relation between bottom-up processing (pixels to higher pixemes) and top-down projections (from semantic entities to pixemes), and sketch a computational model of the visual system that can systematically re-create a visual input in the form of a painting.

A strictly engineering perspective is finally taken in the text of **Hermes** and **the SVP Group** (Germany), which also broadens the view to moving pictures: how can an accept-able movie trailer – as a kind of cinematic summary – be more or less automatically generated on basically syntactic principles from the movie. In the practical point of view, the theoretical discussion about the elements of picture morphology retreats behind the complicated concrete problems at hand. Due to that complexity, the task has even to be restricted to a certain genre (and certainly bound by the current "taste" for trailer esthetics). The focus is mainly on "shots" and the transitions between them. The group presents a program system, the outcome of which has been empirically compared with satisfying results to commercial trailers produced in the ordinary way. In contrast to the neurophysiologically inspired analysis of an input picture in the contribution of du Buf and Rodigues, the input movie is analyzed with several standard techniques from computer vision like motion-based segmentation, and face detection and recognition (supplemented by a range of classification/recognition methods for acoustic input or even text) – techniques that are not necessarily cognitively adequate but basically optimized for the tasks they have to solve. Unlike the system of du Buf and Rodrigues, the final (re)creation of a (moving) picture depends on a separate set of templates following semantic and pragmatic aspects.

As the thematic issue of IMAGE on computational image morphology attempts in particular to mediate between computational visualistics and other disciplines investigating pictures and their uses, a final chapter broadens the perspective again and relates the computational argumentations of the preceding papers to the more general discussion of image science. I, then, also extend the discussion to the question of syntactically ill-formed pictures and the limits of pictorial syntax or morphology.

References

Asher, Nicholas & Vieu, L.: Toward a Geometry of Common Sense: A Semantics and a Complete Axiomatization of Mereotopology. in: IJCAI-95, 846–852, 1995.

Clarke, B.: A calculus of individuals based on 'connection'. *Notre Dame Journal of Formal Logic* 22(3):204–218, 1981.

Ehrig, Hartmut, & Mahr, B.: *Fundamentals of Algebraic Specification*. Berlin: Springer, 1985.

Goodman, Nelson: *Languages of Art. An Approach to a Theory of Symbols*. Indianapolis: Hackett, ^{1}1968, ^{2}1976.

Klee Paul: *Pädagogisches Skizzenbuch*. (Nachdruck: hg. von Hans M. Wingler Berlin: Gebr. Mann, ^{4}1997).

Metzger, Wolfgang: Figurale Wahrnehmung. In: Metzger W. (Ed.): *Handbuch der Psychologie.* Vol. 1, Göttingen: Hogrefe, 1966.

Plümacher Martina: Wohlgeformtheitsbedingungen für Bilder?. In: Sachs-Hombach, K. & Rehkämper, K.: *Bildgrammatik: Interdisziplinäre Forschungen zur Syntax bildlicher Darstellungsformen*. Magdeburg: Scriptum, 1999, 47–56.

Pratt-Hartmann, Ian: Empiricism and Rationalism in Region-based Theories of Space. *Fundamenta Informaticae* 34:1–31, 2000.

Ros, Arno: Was ist Philosophie? In: Richard Raatzsch (ed.): *Philosophieren über Philosophie*. Leipzig: Leipziger Universitätsverlag, 1999, 36–58.

Sachs-Hombach, Klaus: *Das Bild als kommunikatives Medium*. Köln: Herbert von Harlem, 2003.

Sachs-Hombach, Klaus & Schirra, Jörg R.J.: Von der interdisziplinären Grundlagenforschung zur computervisualistischen Anwendung. Die Magdeburger Bemühungen um eine allgemeine Wissenschaft vom Bild. *Magdeburger Wissenschaftsjournal* 1/2002, 27–38.

Sachs-Hombach, K. & Rehkämper, K.: *Bildgrammatik: Interdisziplinäre Forschungen zur Syntax bildlicher Darstellungsformen*. Magdeburg: Scriptum, 1999.

Saint-Martin, Fernande: *Semantics of Visual Language*. Bloomington: Indiana Univ. Press, 1990.

Schirra, Jörg R.J.: *Foundation of Computational Visualistics*. Wiesbaden: DUV, 2005.

Schirra, Jörg R.J. & Sachs-Hombach, Klaus (2006a): Fähigkeiten zum Bild- und Sprachgebrauch. *Deutsche Zeitschrift für Philosophie*. Berlin, Band 54(2006)/6, 887–905.

Schirra, Jörg R.J. & Sachs-Hombach, Klaus (2006b): Bild und Wort. Ein Vergleich aus bildwissenschaftlicher Sicht. *Essener Linguistische Skripte – elektronisch (ELiSe)*, 6(2006)/1:51–72, 2006.

Schlechtweg, Stefan & Raab, A.: Rendering Line Drawings for Illustrative Purposes. In. Strothotte, Th. (ed.): *Computer Visualization – Graphics, Abstraction, and Interactivity*. Springer, Heidelberg, 1998, Chapter 4, 63–89.

Strassmann, S.: Hairy Brushes. In: *Proceedings of SIGGRAPH'86*, 225–232, 1986.

Scholz, Martin: Gestaltungsregeln in der pictorialen Kommunikation. In: Sachs-Hombach, K. & Rehkämper, K.: *Bildgrammatik: Interdisziplinäre Forschungen zur Syntax bildlicher Darstellungsformen*. Magdeburg: Scriptum, 1999, 273–286.

Vieu, Laure: *Sémantique des relations spatiales et inférences spatio-temporelles : Une contribution à l'étude des structures formelles de l'espace en Langage Naturel*. Toulouse: IRIT, Université Paul Sabatier, 1991.

Wittgenstein, Ludwig: *Philosophical Investigations*. New York: MacMillan, 1953.

[Table of Content]

Yuri Engelhardt

Syntactic Structures in Graphics

Abstract

Building upon the existing literature, we are suggesting to regard the building blocks of all graphics as falling into three main categories: a) the graphic *objects* that are shown (e.g. a dot, a pictogram, an arrow), b) the meaningful graphic *spaces* into which these objects are arranged (e.g. a geographic coordinate system, a timeline), and c) the graphic *properties* of these objects (e.g. their colors, their sizes). We suggest that graphic objects come in different *syntactic categories*, such as *nodes*, *labels*, *frames*, *links*, etc. Such syntactic categories of graphic objects can explain the permissible spatial relationships between objects in a graphic representation. In addition, syntactic categories provide a criterion for distinguishing meaningful basic constituents of graphics. Based on the above, we discuss how the concept of *syntactics* can be applied to graphics. Finally we distinguish different types of *meaningful graphic spaces* that can be used to construct graphics. Throughout the paper we relate our proposals to the relevant existing literature.

1 Graphics

Visual displays of information are playing an increasing role in modern society. Think of anything from simple subway maps on the wall, to infographics in the newspaper, to interactive 3D data visualizations on the computer. The focus of this paper is on such diagrams, maps, charts, graphs, tables, and information visualizations. In other words, this paper is not primarily about pictures in the sense of images of physical scenes and objects. Nor is it about art. It is about images that can be regarded as ‘visualizing the non-visual’ in an attempt to clarify information of some sort. Such images are often collectively referred to as “graphics”.

Various scholars have tried to approach graphic representations with concepts from linguistics. Is there such a thing as a “grammar of graphics”? Which level of visual detail is useful for distinguishing basic constituents of graphics? Do constituents of graphics – like constituents of speech – come in different grammatical categories? Building upon the existing literature on these topics, we are trying to answer these questions.

2 Earlier “grammatical approaches” to graphics

Various authors have attempted to approach graphics with the linguistic concept of *grammar*. Let us briefly review a few examples. In 1914, Willard Brinton writes in his book *Graphic methods for presenting facts* that “The principles for a grammar of graphic presentation are so simple that a remarkably small number of rules would be sufficient to give a universal language”. In 1967, Jacques Bertin publishes his classic *Sémiologie graphique*, in which he analyses the “language” of graphic representations and the “visual variables of the image”. In 1976, linguist Ann Harleman Stewart examines the properties of diagrams and claims that “Like any language, graphic representation has a vocabulary and a grammar”. In 1984, Clive Richards proposes a “grammatically-based analysis” of diagrams in his Ph.D. thesis *Diagrammatics*. In 1986, Jock Mackinlay suggests that “graphical presentations are actually sentences of graphical languages that have precise syntactic and semantic definitions”. In Mackinlay’s approach, “the syntax of a graphical language is defined to be a set of well-formed graphical sentences”. In 1987, Fred Lakin publishes his paper “Visual grammars for visual languages”, in which he describes his approach to the “spatial parsing” of graphics, which he defines as “the process of recovering the underlying syntactic structure of a visual communication object from its spatial arrangement”.

Since the mid-nineties the literature on grammatical aspects of graphics is expanding further. Kress and van Leeuwen publish their book *Reading images: the grammar of visual design* (1996). Unfortunately, it is difficult to extract a systematic approach to a syntactic analysis of graphics from their book. A paper titled “The visual grammar of information graphics” (1996) by Engelhardt *et al.*, suggests “syntactic categories of visual components”. Robert Horn, in his book *Visual Language* (1998), proposes a morphology and a syntax of visual language based partly on the work of Jacques Bertin and on the Gestalt principles of perception. In his book *The grammar of graphics* (1999), Leland Wilkinson describes an approach to graphics that is related to object-oriented design in computer science. However, he uses grammatical terminology “metaphorically”, and not in a linguistic sense. Colin Ware (2000) writes about the “perceptual syntax of diagrams”, describing “the grammar of node-link diagrams” and “the grammar of maps”. Engelhardt, in his Ph.D. thesis *The language of graphics* (2002) provides a detailed proposal for the analysis of syntactic structure, which he applies to a broad spectrum of graphic representations. Based on all this previous work, what can we say about the structure of graphics?

3 Building blocks: objects, spaces, properties

To be able to talk about the building blocks of graphics, let us introduce some terminology. We propose a notion of *graphic objects* that will allow for recursive structures: Any graphic representation – *and* any meaningful visible component of a graphic representation – may be referred to as a *graphic object*. This means that graphic objects can be distinguished at various levels of a graphic representation. For example, a map or a chart in its entirety is a graphic object. In addition, the various symbols or components that are positioned within that map or chart are graphic objects as well.

A set of graphic objects can be combined into a meaningful arrangement, together forming a single graphic object at a higher level. As Winn (1991) writes: “One property of the symbol system of maps and diagrams is that their components can form clusters, which in turn can form other clusters in a hierarchical fashion. Each cluster can then act as a discrete component.” Let us give a top-down description of this principle: A *graphic object* (e.g., a map, a time chart) can contain a *graphic space* (e.g., a cartographic space, or the space defined by a time line, see section 7 for more about graphic spaces). In turn, that *graphic space* can contain *graphic objects* (e.g., symbols or small “sub-graphics”). This can be applied recursively, resulting in objects inside spaces inside objects etc. A bottom-up description of this principle was given above: a set of *graphic objects* can be arranged into a *graphic space*, together forming a single *graphic object* at a higher level. This “nesting” or “embedding” (Engelhardt 2002) of graphic structures can be referred to as “recursive composition” (Card 2003). In section 5 of this paper we will come to the question of which level of visual detail is useful for distinguishing basic graphic objects.

In contrast to the general notion of “space”, the notion of *meaningful graphic space* (Engelhardt 1998, 1999, 2002) involves signification: a spatial position stands for something. In many graphics, for example in maps and in time charts, a change of position of an object will correspond to a change of meaning. In technical terms, a meaningful graphic space could be defined as a graphic space that involves an interpretation function from spatial positions to one or more domains of information values. For example, moving to the left in a graphic space may mean moving towards the West (in case of a map), or moving back in time (in case of a time chart). In his paper “Giving meaning to place: Semantic spaces”, Wexelblat (1991) explains that visualizations “give representational significance to arrangement and location”, and that “location may have precise meaning even without the presence of an object at that location”. Card (2003), referring to Engelhardt *et al.* (1996), explains that in a visualization, “Empty space itself, as a container, can be treated as if it had metric structure”. Card presents spatial axes as “an important building block” of graphics.

Before we continue, let us first try to say more about the different categories of “building blocks” of graphics. In graphics, not only the possible constituents themselves (*graphic objects*), and the diverse possible ways of arranging these constituents (in *meaningful graphic spaces*), but also the possible visual appearances of these constituents (*graphic properties* such as size, color), could be considered as being part of the graphic “vocabulary”. In this sense we can say that the building blocks of graphics fall into three main categories: graphic *objects*, meaningful graphic *spaces*, and graphic *properties*. Consider a drawing of a family tree for example. In a family tree, the names and the lines between the names are graphic *objects*. The meaningful graphic *space* into which these graphic objects are arranged involves a vertical ordering of generations (e.g., grandparents on top, grandchildren at the bottom). And if names are written in different colors or sizes, then these are graphic *properties* of those names.

The three categories of the building blocks of graphics – *objects*, *spaces*, and *properties* – can be traced back in the literature, although various different terms have been used to refer to them. Please take a look at table 1. In his classic *Sémiologie graphique*, Jacques Bertin (1967) elaborates on the uses of “marks”, “positional variables”, and “retinal variables”. Twyman’s “schema for the study of graphic language” (1979) sets out “mode of symbolization” against “method of configuration”. Wexelblat (1991) describes visualizations as “represented objects” that are positioned in “semantic spaces”. Winn (1991) dissects maps and diagrams into “components” and their “configuration”. Engelhardt *et al.* (1996) distinguish “visual components”, “basic operations of spatial syntax”, and “visual appearance”. Card, Mackinlay and Shneiderman (1999) and Card (2003), both referring to Engelhardt *et al.* (1996), introduce the term “spatial substrate”.

Meaningful graphic spaces are elaborated on in section 7. In the next two sections we will examine graphic objects.

Building blocks *of graphics*	*graphic* ***objects***	*graphic* ***spaces***	*graphic* ***properties***
Bertin (1967)	marks	positional variables	retinal variables
Twyman (1979)	mode of symbolization	method of configuration	—
Wexelblat (1991)	represented objects	semantic spaces	—
Winn (1991)	components	configuration	—
Engelhardt *et al.* (1996)	visual components	basic operations of spatial syntax	visual appearance
Card *et al.* (1999), Card (2003)	marks	spatial substrate	retinal properties

Table 1: The building blocks of graphics

4 Syntactic categories of graphic objects

Every known spoken and/or written language is based on the possibility of combining language constituents of different *syntactic categories*. Examples of such syntactic categories are “noun phrase” and “verb phrase” (sometimes referred to as phrasal categories), or “noun”, “verb” and “adjective” (usually referred to as lexical categories, or ‘parts of speech’). In natural languages, such syntactic categories usually differ from each not only with regard to syntactic aspects but also with regard to semantic aspects.

Graphics can be approached in a similar way. Richards (1984) provides a very simple example figure in which a letter “A” and a letter “B” are connected by a line. This figure represents visually that “A is connected to B”. Richards suggests that in this case “the line serves a verb-like function for the nouns A and B”. A different way of describing this is to say that this simple figure contains *nodes* (the letters “A” and “B”) and a *connector* (the line connects “A” and “B”). Mackinlay (1986) uses the term “connection languages” and writes that “Sentences of connection languages consist of two sets of marks: the set of nodes [...] and the set of links [...]” (again, in our terminology: a set of *nodes* and a set of *connectors*). As Mackinlay points out, it is also syntactically relevant here that “The nodes constrain the position of links”.

To make a more general statement, we claim that all graphics are based on the possibility of combining graphic constituents (graphic objects) of different *syntactic categories* (Engelhardt *et al.* 1996, Engelhardt 2002, 2006). Let us take a subway map as an example. On a subway map, each subway station is indicated by a graphic object (e.g., a dot, or a small circle, or a tick). In terms of our analysis, that graphic mark functions syntactically as a *node*. Next to that graphic mark we read the name of that particular subway station. That station name functions syntactically as a *label*. The paths taken by different subway lines are represented as lines of different color. These colored line segments between the subway stations function syntactically as *connectors*.

These three syntactic categories reflect the existence of discrete entities (*nodes*), their specification (*labels*), and their connections (*connectors*). While *nodes* may make sense by themselves (icons for example), *labels* and *connectors* only make sense in the presence of the *nodes* that they are labeling or connecting (Engelhardt *et al*. 1996). By the way, a label does not need to be textual. In the London Underground map for example, subway stations that are close a train station are labeled with the *graphic symbol* that stands for the British Railways.

These syntactic categories may apply to a subway map, but how about a topographic map? Well, a topographic map may for example contain red dots that function as *nodes* indicating cities. In addition, the topographic map may contain blue lines that function as *line locators* indicating rivers; and, for example, small blue areas that function as *surface* *locators* indicating lakes. And the map may contain words that function as *labels* naming all these cities, rivers and lakes.

Graphic objects of different syntactic categories “behave” differently in a graphic representation. The constraints that govern their spatial positioning are different. Let us look at three examples. Example 1: What makes a *connector* different from a *line locator*? Consider a map that shows airline services between cities. Such a map will usually use *connectors* to show which cities are connected by flights. A *connector* is attached to the two graphic objects that are connected by it, and can easily been drawn with a slightly different curve, possibly making it bend a little more in order to prevent it from running through the middle of a third city in between, for example. A *line locator* on the other hand, such as a blue line that indicates a river on that same map, is attached to every point along the line that is described by the course of that river. The mapmaker can (should) not, for example, bend the line a little, in order to prevent it from running through a certain city. The reason for this is that this line is not simply a *connector* that links spring to ocean, but a *line locator* that traces a specific line in space.

Example 2: What makes a *label* different from a *node*? Consider a small black square on a map that indicates the location of a city, with a word indicating the name of the city (e.g., “Amsterdam”). That word is a *label*, which is attached to the black square. If more convenient for some reason, the mapmaker can move the label to the other side of the black square, as long as the label remains close to the black square. The black square however is a *node*, which is attached to a point in graphic space. This means that, while the mapmaker can move the label to the other side of black square, he cannot move the black square to the other side of the label. (In the latter case he would be moving the city.)

Example 3: What makes a *node* different from a *surface locator*? Consider two colored shapes on a map. One of the colored shapes is a pictogram of some sort that indicates a particular location (e.g., “you are here”). The other colored shape indicates a lake. The first colored shape (“you are here”) is a *node*, which is attached to a point in graphic space. This colored shape can be made somewhat bigger or smaller by the mapmaker, without a change in meaning. The second colored shape (lake) is a *surface locator*, which is attached to a specific surface in graphic space. Consequently, the mapmaker cannot, for example, make this colored shape somewhat bigger or smaller without a change in meaning. (The lake would grow or shrink.)

Nodes, *labels*, *connectors*, *line locators*, and *surface locators* are examples of frequently used syntactic categories of graphic objects. *Proportional segments* are an example of a syntactic category that appears specifically in pie charts (the pie segments), in stacked bar charts, and more recently, in “treemaps”. See table 2 for a few more syntactic categories.

Syntactic categories of graphic objects:	**Type of attachment:**	**Example(s):**
node	is attached to: either a point in a meaningful graphic space, or to nothing	a dot marking a city on a map, or a text box in a flow chart
label	is attached to: a graphic object that is labelled by it	a name labelling an object on a map
connector	is attached to: two graphic objects that are connected by it	a line connecting two names in a family tree
line locator	is attached to: a specific line in a meaningful graphic space	a river on a map, or the curve of an electrocardiogram
surface locator	is attached to: a specific surface in a meaningful graphic space	a colored surface on a map, representing a lake or a country
grid marker	is attached to: points and lines of orientation in a meaningful graphic space	latitude/longitude lines on a map, or axes and tick marks in a chart
proportional segment	is attached to: a segment of the surface of a graphic object	a pie segment in a pie chart
frame	is attached to: the graphic object that is framed by it	the line around the panel in a comic book
etc....	etc....	etc....

Table 2: Syntactic categories of graphic objects and rules for their combination.

Corresponding to "parts of speech" in natural languages, one could refer to these syntactic categories in graphics as "graphic parts".

All syntactic categories of graphic objects can be divided into two main groups: 1) objects that are attached to locations in graphic space (e.g., *node, line locator, surface locator, grid marker* are all attached to locations in graphic space), and 2) objects that are attached to other objects (*label, connector, proportional segment, frame* are all attached to other objects).

Several of the examples we that we are using above are taken from maps. However, we claim that *all* types of graphic representation of information can be analyzed in terms of their composition from graphic objects of different syntactic categories. For a more complete list of syntactic categories see “The language of graphics” (Engelhardt 2002) and “Objects and spaces: The visual language of graphics” (Engelhardt 2006).

5 At which level of detail do we define basic graphic objects?

If we wish to regard graphics as sign systems, at which level of visual detail should we look for the ‘basic signs’ that graphics are composed of? What is the lowest level at which it is useful to talk about *graphic objects* in the sense of the approach that is proposed here? Richards (1984) believes that “there seems to be little profit in using such items as an individual dot or line as a unit of analysis. If we are going to use linguistics as a model, then what is needed for present purposes is not the pictorial equivalent of a phoneme or morpheme but something closer to a noun phrase”. A little further on, Richards formulates it even stronger: “If any analysis is going to be possible at all it seems that it must start at a ‘noun phrase’ level, otherwise we are forced down to the meaningless level of dots and lines or else up to the level where all we can say is ‘here is a diagram’“.

How would “a ‘noun phrase’ level” generalize to the approach that is proposed here? Well, it points us to the (lowest) levels at which syntactic categories of graphic objects can be observed. This leads us to the following proposal:

> The *basic graphic objects* in a particular graphic representation are those that can be regarded as functioning in some *syntactic category* within that particular graphic representation (e.g., as a label, as a node, as a connector, as a proportional segment, etc.).

In other words, we use the term *basic graphic object* to mean the smallest visual entities that play some syntactic role in the sense that we have been discussing in the previous section. Having explored syntactic aspects of graphic objects, we will now take a look at how the concept of *syntactics* can be applied to graphics.

6 Syntactics

The distinction between syntactics, semantics, and pragmatics was introduced by Charles Morris (1938, 1946). Morris conceives of *syntactics* as the investigation of the relationships between signs, of the ways in which complex signs can be constructed from simple ones, as well as the ways in which complex signs can be analyzed into more simple ones (Morris 1946/1971). MacEachren (1995) notes:

> According to Morris, syntactics is the relation between a given sign vehicle and other sign vehicles. There is a critical distinction here (that many cartographers have missed) between Morris’s “syntactics” and the linguistic subcategory of “syntax”. While syntax puts emphasis on word order and parsing (i.e., on a linear sequence), syntactics is much

> broader in scope. Syntactics allows for any kind of among-sign relationships. Morris (1938, p. 16) makes this point explicitly in his statement that there are “syntactical problems in the fields of perceptual signs, aesthetic signs, the practical use of signs, and general linguistics.” [...] At least three kinds of sign relationships seem to fall under Morris’s umbrella of syntactics (Posner, 1985, in French; cited in Nöth, 1990, p. 51). These include: (1) “the consideration of signs and sign combinations so far as they are subject of syntactical rules” (Morris, 1938, p. 14), (2) “the way in which signs of various classes are combined to form compound signs” (Morris, 1946/1971, p. 367), and (3) “the formal relations of signs to one another” (Morris, 1938, p. 6).
>
> [MacEachren 1995]

All of the descriptions of syntactics given above fit perfectly with the approach to graphic structure that is proposed in this paper. The syntactics of graphics investigates the relationships between graphic objects of different syntactic categories. It investigates the rule- and constraint-based relationships between graphic objects (of different syntactic categories) and graphic spaces. And syntactics investigates how graphic objects can be combined into composite graphic objects, and how composite graphic objects can be analyzed into more simple ones.

So far, we have concentrated on the discussion of graphic *objects*. The uses of graphic *properties* (e.g., color, size) have been thoroughly investigated by Bertin (1967), and later, among others, by Mackinlay (1986) and by MacEachren (1995). We will now take a closer look at the third main category of the building blocks of graphics: meaningful graphic *spaces*.

7 Meaningful graphic spaces

Imagine sitting in a bar and using the arrangement of empty beer glasses on the bar table to explain, say, the location of Berlin with respect to London and Paris. The positioning of two beer glasses, standing for London and Paris, creates a *meaningful space* (Engelhardt 1998, 1999, 2002) – every position on the bar table has been assigned a geographical meaning. The meaningful space can even be regarded as extending beyond the bar table – a person on the other side of the bar may now happen to be “sitting in Africa”. Similarly, when starting to draw a financial chart, by drawing two labeled axes (e.g., one for the months of the year, and the other for expenses in dollars), a *meaningful graphic space* has been created: every position in the yet-empty chart has been assigned a meaning, even before we have any data. The face of a clock also constitutes a meaningful graphic space - it assigns meaning (time of day) to every spatial position along a circle. While the “London-Paris-Berlin space” represents a *physical* space, the empty financial chart and the clock face represent a *conceptual* space. This is a pretty straightforward but important distinction.

Looking at the broad spectrum of graphics we can say that images of physical scenes and objects, such as pictures and maps, represent *physical spaces*, while many abstract graphics, such as family trees and statistical charts, represent *conceptual spaces* (Engelhardt 1999, 2002). In other words, pictures and maps use *spatial* arrangement in the image to represent *spatial* arrangement in the world, while family trees and pie charts use *spatial* arrangement in the image to represent *non-spatial* information.

Let us take a brief look at the relevant terminology in the literature. Regarding the frequently used term “iconic”, we can assert that “iconic” graphics (such as pictures and maps) display *physical* spaces, while “abstract” graphics (such as family trees and statistical charts) display *conceptual* spaces. The former represent “concrete objects”, while the latter represent “intangible concepts” (Winn 1991). This distinction has also been referred to as portraying “visible things” versus portraying “things that are inherently not visible” (Tversky 2001). One could argue however, that some representations of physical spaces such as a drawing of a molecule, a floor plan, or a world map, are – strictly speaking - not portraying “visible things”. Therefore, instead of “visible” versus “non-visible”, the distinction between *physical* and *conceptual* seems more appropriate here. Accordingly, we can observe that some (aspects of) graphics are “meant to reflect physical reality” while other (aspects of) graphics are “meant to reflect conceptual reality” (Tversky 2002).

Representations of physical spaces do, by the way, not always have to express the true co-ordinate proportions of the represented objects. Think of a world map, the London Underground map, or an “exploded view” of a machine. All three of these images greatly distort the physical spaces that they show, but nevertheless they are still representations of physical spaces.

Many graphics combine physical and conceptual spaces. As an example, think of little pictures of people or things (showing *physical* spaces) that are arranged on a time line (representing a *conceptual* space). Richards (1984) points out that while the “perspective landscape is homogeneous in that it portrays a single unbroken space at a single moment in time [...] it seems that more than one space and more than one time can be portrayed in a single diagram”. As another example, think of little bar charts (showing *conceptual* spaces) that are arranged on a map (showing a *physical* space). Both of these examples make use of what can be referred to as “nesting”, “embedding” (Engelhardt 2002), or “recursive composition” (Card 2003).

As an example of a true *hybrid* space (Engelhardt 1999, 2002), think of a three-dimensional landscape drawing of a country in which the drawn “mountains” do not represent physical mountains, but – for example - population density, peaking in the cities and flat in the countryside. In this case, the horizontal plane represents the *physical* space of the country’s geography, while the vertical dimension represents the *conceptual* space of

A typology of meaningful graphic space: Alternative terminology and explanations:		*representation of* ***physical space***	*representation of* ***conceptual space***
“proportion” (in French, Bertin 1967) “interval” (Tversky 1995) “quantitative” (Engelhardt *et al.* 1996) “ratios of spatial distances [...] are perceived as meaningful” (Engelhardt 2002) “quantitative grid” (Card 2003)	**metric space** (shows proportions)	e.g., a topographic map, most pictures	e.g., a time axis, any other quantitative axis
“ordre” (in French, Bertin 1967) “ordinal” (Tversky 1995, Engelhardt *et al.* 1996) “a metric space that was printed on a ‘rubber sheet’ and then stretched non-homogenously” (Engelhardt 2002) “ordinal grid” (Card 2003)	**topological space** (shows order)	e.g., the London Underground map, an “exploded view” of a machine	e.g., chronological ordering of panels in a comic, any other meaningful spatial ordering
“association” (in French, Bertin 1967) “categorical” (Tversky 1995, Engelhardt *et al.* 1996) “segmentation” (Engelhardt 1998, 1999) “spatial clustering” (Engelhardt 2002) “nominal grid” (Card 2003)	**grouping space** (shows association)	e.g., columns and rows in a table, any other meaningful spatial grouping	
a) “recursion is the repeated subdivision of space” (Card *et al.* 1999) “nesting”, “embedding” b) (Engelhardt 2002) b) “orthogonal placement of axes” (Card *et al.* 1999) “simultaneous combination” (Engelhardt 2002)	**composite space** (constructed from combinations of the spaces above)	a) e.g., the (metric, physical) space of a picture within the (topological, conceptual) space of a chronological sequence, b) e.g., a chart that combines a (metric, conceptual) horizontal time axis with a (metric, conceptual) vertical quantitative axis	

Table 3: A typology of meaningful graphic space

population density. This example makes use of what can be referred to as “orthogonal placement of axes” (Card *et al.* 1999) or “simultaneous combination” (Engelhardt 2002).

In table 3, the two main operations for combining basic graphic spaces into composite graphic spaces are marked as “a” (embedding) and “b” (orthogonal placement of axes). These two techniques could be regarded as “composition operators that can generate composite designs” (Mackinlay 1986).

Not all graphic spaces are meaningful graphic spaces by the way. A set of graphic objects can also be shown in a random arrangement (Engelhardt *et al.* 1996), forming a more or less arbitrary spatial structure (Engelhardt 2002). In this case the involved graphic space is "unstructured" (Card *et al.* 1999, Card 2003).

We claim that *all* types of graphic representation of information can be analyzed in terms of their composition from graphic spaces of different sorts. For a more complete discussion of meaningful graphic spaces see Engelhardt 2002 and Engelhardt 2006.

8 Conclusions

Graphics can be regarded as expressions in visual languages. We have tried to show that specifying such a visual language means a) specifying the syntactic categories of its *graphic objects*, plus b) specifying the *graphic space* in which these graphic objects are positioned, plus c) specifying the visual coding rules that determine the *graphic properties* of these graphic objects (see table 1). The *syntactic structure* of a graphic representation is determined by the rules of attachment for each of the involved syntactic categories (see table 2) and by the structure of the meaningful graphic space that is involved (see table 3). With this analysis we have attempted to demonstrate that Morris' original notion of *syntactics* applies well to the structure of graphics.

References

Bertin, J.: *Sémiologie graphique*. Paris: Editions Gauthier-Villars. English translation by W.J. Berg (1983) as Semiology of graphics, Madison, WI: University of Wisconsin Press, 1967.

Brinton, W.C.: *Graphic methods for presenting facts*. New York: McGraw-Hill Book Company / The Engineering Magazine Company, 1914.

Card, S.K., J.D. Mackinlay, and B. Shneiderman (Eds.): *Readings in information visualization: Using vision to think*. San Francisco: Morgan Stanley Kaufmann Publishers, 1999.

Card, S.K.: Information visualization. In: J.A. Jacko and A. Sears (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Application*. Mahwah, New Jersey: Lawrence Erlbaum Associates, 2003, 544–582.

Engelhardt, Y., J. de Bruin, T. Janssen, and R. Scha: The visual grammar of information graphics. In: N. H. Narayanan, and J. Damski (Eds.), *Proceedings of the AID '96 Workshop on Visual Representation, Reasoning and Interaction in Design*, Stanford University, 1996.

Engelhardt, Y.: Meaningful space: How graphics use space to convey information. In: *Proceedings of Vision Plus 4, School of Design*, Carnegie Mellon University, Pittsburgh, 1998, 108-126.

Engelhardt, Y.: Meaningful space. In: *If/Then: Design implications of new media*, Netherlands Design Institute, 1999, 72–74.

Engelhardt, Y.: *The language of graphics: A framework for the analysis of syntax and meaning in maps, charts and diagrams*. Unpublished Ph.D. thesis, Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands, 2002.

Engelhardt, Y.: Objects and spaces: The visual language of graphics. In: D. Barker-Plummer, R. Cox, and N. Swoboda (Eds.) *Diagrammatic representation and inference*. Berlin: Springer, 2006.

Horn, R.E.: *Visual language: Global communication for the 21st century*. Bainbridge Island, WA: MacroVU, Inc., 1998.

Knowlton, J.Q.: On the definition of "picture". *AV Communication Review*, 14, 1966, 157–183.

Kress, G., and T. van Leeuwen: *Reading Images: The Grammar of Visual Design*. London: Routledge, 1996.

Lakin, F.: Visual grammars for visual languages. In: *Proceedings of AAAI 87, Sixth National Conference on Artificial Intelligence*, Seattle, Washington, 1987, 683–688.

MacEachren, A.M.: *How maps work: Representation, visualization, and design*. Guilford Press, 1995.

Mackinlay, J.D.: Automating the design of graphical presentations. *ACM Transactions on Graphics* 5, 2, 1986, 110–141.

Morris, C.W.: *Foundations of the Theory of Signs: Foundations of the Unity of Science: Towards an International Encyclopedia of Unified Science*, Volumes 1 and 2. Chicago: University of Chicago Press, 1938.

Morris, C.W.: Signs, language and behavior. In C. Morris (Ed.) (1971), *Writings on the General Theory of Signs*. The Hague: Mouton, 1971 (originally published in 1946).

Nöth, W.: *Handbook of Semiotics*. Bloomington: Indiana University Press, 1990.

Richards, C.J.: *Diagrammatics: An investigation aimed at providing a theoretical framework for studying diagrams and for establishing a taxonomy of their fundamental modes of graphic organization*. Ph.D. thesis, Royal College of Art, London, 1984.

Stewart, A.H.: *Graphic representation of models in linguistic theory*. Bloomington: Indiana University Press, 1976.

Tversky, B.: Cognitive origins of graphic conventions. In: F.T. Marchese (Ed.), *Understanding images*. Berlin, New York: Springer, 1995, 29–53.

Tversky, B.: Spatial schemas in depictions. In: M. Gattis (Ed.), *Spatial schemas and abstract thought*. Cambridge: MIT Press; 2001.

Tversky, B.: Some ways that graphics communicate. In N. Allen (Ed.), *Working with words and images: New Steps in an Old Dance*. Westport, Connecticut: Greenwood, 2002.

Twyman, M.: A schema for the study of graphic language. In: P.A. Kolers, M.E. Wrolstad, H. Bouma (Eds.). *Processing of visible language*. Vol. 1., 117-150. New York: Plenum Press, 1979.

Wexelblat, A.: Giving meaning to place: semantic spaces. In: M. Benedikt (Ed.), Cyberspace: first steps, Cambridge, MA: MIT Press, 1991, 255–271.

Wilkinson, L.: *The Grammar of graphics*. Berlin, New York: Springer, 1999.

Winn, W.: Learning from Maps and Diagrams. *Educational Psychology Review*, Vol. 3 (1991), No. 3, 211- 247.

[Table of Content]

Stefano Borgo, Roberta Ferrario, Claudio Masolo and Alessandro Oltramari

Mereogeometry and Pictorial Morphology

Abstract

The paper reviews geometrical approaches in the area of qualitative space representation by discussing formal systems of geometry based on the notion of extended regions (mereogeometries). The focus is on primitives that are cognitively motivated and that capture different notions of naive geometry. The paper then moves to consider the role of mereogeometries (and in particular of the concepts they rely upon) in the domain of picture morphology in two ways: it discusses some primitives that are motivated from the cognitive perspective, and it considers the issue of granularity and refinement.

1 Introduction

Space, as the realm of physical locations or as the structure where to organize knowledge, has always been at the center of scientific research. In this work we look at space in two distinct senses: on the one hand the characterization of physical space as provided by mereogeometries [Borgo & Masolo *to appear*], on the other the study of space in the light of pictorial representation and, more specifically, of pictorial morphology [Schirra 2005]. If mereogeometries are the result of a formal approach to geometry that was primarily developed in the 20th century and that tries to do justices of cognitive and foundational principles, pictorial morphology is the research area where images are analyzed and decomposed with tools inspired by techniques developed in linguistics (e.g., generative grammars). The two approaches are fairly recent and their possibility of interaction seems strong. The paper first reviews the motivations and the development of mereogeometries and then moves to investigate the relationships between mereological primitives and research in pictorial morphology. The goal is not to set a precise comparison, which would be premature since the connection between these areas is still in a primitive state, but to look at commonalities and to suggest possible future investigations in the respect of the particularity and aims of each discipline.

2 Mereogeometry

Mereogeometry is a form of geometry (that is, a mathematical theory) that has found its place not in the usual mathematical community but within the knowledge representation area and, more precisely, in the domain of qualitative representation. The interest on representation of space based on mereology (i.e., the relation of parthood over extended regions) goes back at least to Lobacevskij [1835] but it has not catch the attention of a consistent number of researchers before the end of the last century. From the 90s, research on this topic has finally become consistent and the number of papers devoted to this area has regularly increased since then.

Going back to what we may consider the beginning of research in mereogeometry, we find Lobacevskij's work (published in 1835) where the author posits as task of his research the search of an alternative to the axiomatic foundation of geometry based on points. At the time, geometry was by antonomasia Euclidean geometry. However, some researchers were pointing out that this system falls short of satisfying cognitive concerns being based on the cognitively disputable notion of point. Indeed, human experience of space is experience in magnitude and points cannot be empirically experienced. Nonetheless, what should be taken as ground for a cognitively and philosophically sound geometrical system and what properties such a system should have was not clear yet. Taking solids as basic entities for his system, Lobacevskij revolutionizes the foundations of geometry from the ontological viewpoint and begins a new field to fill the gap between geometrical and spatial entities.

Little by little, systems of mereogeometry (although not yet called in this way) started to be introduced and discussed with particular emphasis on cognitive soundness and expressiveness. As one could expect, at the beginning the aim was to show that the concept of point is not necessary for the foundation of geometry. After all, the standard approach at the time was to define regions as sets of points. This topic pervades the works of Whitehead [1929], De Laguna [1922], Nicod [1924], Tarski [1956], and Grzegorczyk [1960]. With the introduction of formal techniques to reconstruct points from extended regions, the different conceptualizations of space that were proposed in those years could be seen from a more rigorous perspective. Now the attention was driven to the properties of space, the primitive relations, and the ontological nature of the entities in the adopted domain of discourse. In particular, these authors talk of apparently different entities like solids, extended regions, bodies, and volumes. In some cases these notions are used just informally, and it is difficult to understand the presuppositions or the basic intuitions about (physical) objects and their possible locations in space. In addition, some authors have developed mixed theories where the domain of discourse includes entities of different dimensions like points, lines, surfaces, and volumes (Gotts [1996] and Galton [2004]).

While classical geometry had already been defeated as 'the geometry of physical space' after the introduction of relativist physics, the centrality of Euclidean geometry was now substantially questioned even at the level where it is most successful: the layout of our everyday space.[1] The revolution was brought by the definition of points as particular sets of regions. Since the new theories succeeded in defining Euclidean entities and relations within a different domain, one cannot rely on purely formal arguments to establish which entities and relations deserve the role of geometrical primitives. Euclidean geometry is challenged to defend itself on the choice of the basic entities, an issue that has always been avoided by pointing at the successful history of the discipline in modeling space intuitions.

But now mereogeometries have reached a level of formal clearness and are supported by arguments that arise in the new studies of the relationship between humans, their perceptual and cognitive apparata, and their experienced knowledge of space. Here, region-based geometries seem to be cognitively more appealing since they make possible an (almost) direct mapping from empirical entities and laws to theoretical entities and formulas. At the same time, the new entities are openly discussed: the consequences of choosing extended regions as primitive entities, the meaning of empirically experiencing extended regions, the role of perfect regions in geometrical construction.

It has been with the work of Clarke [1981, 1985] that theories based on extended entities have shown their potentialities for both their formal aspects and their possible use in application. Furthermore, the ontological clearness and the evident connection with physical entities justify the interest of philosophers. The relations of parthood (Greek meros = part, hence *mereology*) and connection (*topology*) are here taken to be fundamental notions exemplified by spatial or material entities like physical objects, chunks of matter, holes, etc. (see Simons [1987], Casati and Varzi [1999], and Smith [1998]). Nowadays, these theories are known as mereotopologies. Then, we can look at mereogeometries as theories that extend mereotopologies with predicates and/or relations of geometrical import. They may be motivated from different research domains, as it will be explained below, since the general idea is to reconstruct a commonsense notion of space as it is understood in those domains.

---

[1] Riemann and Lobacevskij showed that there is no strong evidence in favour of grounding Euclidean geometry on human mental structures, moving away from the view in *Kritik der Reinen Vernunft* (see Kant [1787]). Thus, the notion of an absolute space independent of physical bodies should be discarded, denying also the existence of a pure spatial intuition. In other words, although Euclidean geometry provides a suitable framework to represent the sensible universe, it does not follow that the axiomatic system underlying that geometry has to be thought as embedded in the structure of space. "*Geometry, therefore, so far as it seeks to be a science of space, is by no means independent of physical experiences; and hence […] it does not investigate some sort of "pure space", but rather describes certain aspects of the behaviour of bodies in nature"* (Schlick [1925], pp. 48–49).

Although space has been traditionally captured by point-based geometry, it must be recognized that, overall, the properties of Euclidean space fit our commonsense notion of space. Thus, it should not be surprising that most mereogeometries lead to systems 'equivalent' to Euclidean geometry [Borgo & Masolo, *to appear*]. This very fact shows how our cognitive perception of space is quite stable and precise and is not affected by the choice of geometric primitives. Indeed, the properties that commonsense space should satisfy are not an issue. The crucial point is how we *cognitively* attain this specific notion of space. In this perspective, the first question that mereogeometries try to answer is what primitives apply to extended objects and are expressive enough to generate the commonsense notion of space.

Mereogeometries naturally arise in various areas. In Schmidt [1979] physics is presented as a theory based on extended entities. This theory allows us to refer explicitly to the objects involved in experiments. Generally speaking, cognitive science and computational linguistics analyze the possibility of formalizing human learning, conceptualization, and categorization of spatial entities and relations. In particular, Renz *et al.* [2000] take into account the cognitive adequacy of topological relations while Aurnague *et al.* [1997] and Muller [1998] show how mereogeometrical notions are central in the semantics of natural language. Donnelly [2001] formalizes the theory of De Laguna in the perspective of commonsense analysis of spatial concepts. In computer science and more specifically in qualitative spatial representation and reasoning,[2] mereogeometries are applied for modeling qualitative morphology and movement of physical bodies,[3] for describing geographical spaces and entities in Geographical Information Systems,[4] as well as for characterizing medical and biological information.[5]

In all these areas, specific foundational and applicative concerns affect the development of the theories based on geometrically extended entities. Indeed, in the literature there are numerous mereogeometries that differ on primitive entities, formal properties, as well as general principles. Unfortunately, due to technical difficulties, there are just a few formal studies on the relationships among mereogeometrical systems. In particular the poor axiomatization of most mereogeometries and the lack of a general methodology further complicate the task. A more systematic comparison is encouraged to facilitate both reuse and communication among applications based on different systems.

---

[2] See Cohn & Hazarika [2001] and Vieu [1997] for good overviews.

[3] Bennett *et al*. [2000a, 2000b, 2001], Borgo *et al*. [1996], Cristani *et al*. [2000], Dugat *et al*. [1999], Muller [1998], Galton [2000], Li *et al*. [2003], Randell *et al.* [1989, 1992].

[4] Pratt-Hartmann & Lemon [1997], Pratt-Hartmann & Schoop [2000], Stock [1997].

[5] Schulz [2001], Cohn [2001], Smith & Varzi [1999], Donnelly [2004].

Basic Terminology

The study of space is made possible by adopting a few well-defined concepts. Although in this paper we do not need to look at the precise formal definitions (we will present them in simplified terms), it is always good to try to understand the details and the ontological meaning of a defined concept. The interested reader can look at [Simons 1987] for a more in depth analysis of mereotopological terms and [Borgo & Masolo *to appear*] for the mereogeometrical terms.

At the basis of topology we have the notion of *open set*. An open set is a set that does not contain its boundary: examples are (0, 5) in the one-dimensional space $\Re^1$ (the real line) and $\{(x, y) \mid x^2 + y^2 < 1\}$ in the two-dimensional space $\Re^2$. The dual notion is that of *closed set*, that is, a set that contains its boundary like [0, 5] in $\Re^1$ and $\{(x, y) \mid x^2 + y^2 \leq 1\}$ in $\Re^2$. For a physical example, think of an apple with an extremely thin (we would say 'infinitely thin') skin: the apple without the skin fills an open set, with the skin a closed set. In general, given an open set *A*, the corresponding closed set is the smallest closed set *B* that contains *A* (in turn, *A* is the biggest open set contained in *B*). The *closure* operator highlights this relationship: given an open set *A* as before, the closure of *A* is the set *B*. The difference between an open set and its corresponding closed set is called the boundary. Then, for any set *C*, *C* plus its own boundary is closed (indicated by [C]) while what remains of *C* after its boundary has been dropped is an open set (indicated by C°). Note that the empty set and the universe of domain have no boundary and thus are at the same time open and closed.

A *regular set A* is a set stable under the operations of topological closure (i.e., [ ] and its dual °) in the sense that: (*i*) the closure of a regular set *A* is equal to the closure of the corresponding open set $A°$ (formally $[A] = [A°]$) and (*ii*) the open set of a regular set *A* is equal to the open set of its closure [*A*] (formally $A° = [A]°$). These regions are dimensionally homogeneous in the sense that the conditions exclude objects of mixed dimensions. For example, in $\Re^3$ a solid cube with a point removed or a solid cube with an external segment attached to it are not regular. Finally, since a regular set may be neither open nor closed, an open regular set is a regular set that is also open (analogously for closed regular sets).

Informally, two sets are said to be *connected* when 'they touch each other'. There are several ways to make precise this notion. Below we use it in the following sense: given two non-empty sets *A* and *B*, they are connected if [*A*] and [*B*] (i.e., their closures) share at least one point.

The notion of *self-connection* is introduced to talk about sets that are not scattered; they are 'single pieces' so to speak. A set *A* is *self-connected* if it is impossible to split *A* in two non-empty sets without generating a new extended boundary in each of the parts. The intuition behind this notion is easily grasped when we look at physical objects. Take a chocolate bar: if we image to cut the bar in two parts, we know that each part will present some 'extended' new boundary, namely where the cut takes place ('extended' because it is not just a point or a line; it is a new piece of surface). Compare this with a set of candies: we can split the candies in two groups (dividing them by color or brand or flavor) without generating any new boundary.

The notion of *congruence* captures the idea of objects of same shape and same size. If two objects are congruent, each can fill up the same location that the other does. Relatively to abstract geometrical entities, these notions can be rephrased as: two sets of points are congruent if it is possible to move one over the other (or over a symmetric image of the other) so that each point of the first is co-located with a point of the second and *vice versa*. Note that the movement must be 'rigid', that is, in the movement to fit the other set, no part of the geometrical entity must undergo squeezing or stretching.

3 Mereogeometries, Primitives and Interpretations

The formal interpretation of the non-logical primitives is crucial to understand the expressiveness and the cognitive plausibility of a logical system. For instance, researchers have been interpreting the notion of 'extended region' using different sets of geometrical loci. Common to most approaches is the interpretation of extended regions as regular sets in the space $\Re^n$ (where n is the dimension of the space one is modeling). However, rarely all regular sets are considered; often one restricts the interpretation to the subclass of open regular sets, closed regular sets, polygonal regular sets, finite regular sets, and so on.

Unfortunately, especially in the early works in mereology, this aspect has been mostly neglected. However, to be honest, even in recent literature it may happen that the formal interpretation of the primitives is not addressed. Indeed, sometimes researchers relay on intuitive interpretations and focus just on formal and implementation properties of the primitives. In these cases, the satisfaction of interesting properties is considered as the preliminary condition that may motivate a subsequent logical formalization. In other approaches, the goal of the research is limited to the construction of computationally efficient systems, and, in these cases, the logical formalization is not even attempted.

If one wants to consistently analyze and make a comparative study of these mereogeometrical systems, one needs to start with a general discussion on implicit assumptions which motivate the intended interpretation of the non-logical vocabulary and the adopted domain of discourse. Without this analysis, we suspect, it is unrealistic to look for a suitable (or even correct) framework for a comparison. An example in this sense is carried out in [Borgo & Masolo, *to appear*]. This work discusses and compares some systems of mereogeometry based on different primitives and different domains that we report here.

T1: We begin with the mereogeometry presented in [Tarski 1956] and further developed in [Bennett 2001, Bennett *et al.* 2000]. Here there are two primitives: the binary relation *P* of **parthood** (*P(x, y)* stands for “region *x* is part of region *y*”) and the predicate *S* of **sphere** (*S(x)* stands for “region *x* is a sphere”). The theory has been developed for the domain of non-empty regular open subsets of $\mathfrak{R}^n$. The idea is that *P* can be interpreted as set-inclusion among regions of points in $\mathfrak{R}^n$ and that *S* corresponds to the notion of ball in $\mathfrak{R}^n$.[6]

Figure 1: x *is part of* y, P(x, y)*;* z *is a sphere,* S(z)

T2: This theory was presented in [Borgo & Masolo, *to appear*] and adopts three primitives: *P*, *SR*, *CG*. The first is the relation of **parthood** we have seen in T1. *SR* is the predicate of **self-connectedness**: *SR(x)* is read “region *x* is self-connected” (see Fig. 2). Finally, *GC* is the binary relation of **congruence**: CG(x, y) stand for “regions *x*, *y* are congruent”. The domain for the theory is given by the non-empty regular open subsets of $\mathfrak{R}^n$ with finite diameter. That is, compared to T1, the theory discharges infinite regions.

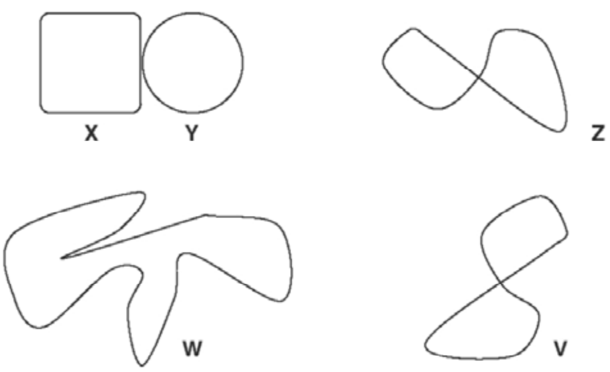


Figure 2: x *and* y *are connected,* C(x, y)*;* w *is self-connected,* SR(w)*, while* z *and* v *are not;* z *and* v *are congruent,* CG(z, v)

T3: The third system was given in [Nicod 1924] and is based on the primitives *P* and *Conj*. As before, *P* is the relation of **parthood**. *Conj* is the quaternary relation of **conjugateness**: *Conj(x, y, z,w)* stands for “regions *x, y* and *z, w* are conjugate”. Informally, this means that there is a point[7] p_x in *x*, a point p_y in *y*, a point p_z in *z* and a point p_w in *w* such that the distance between p_x and p_y equals the distance between p_z and p_w (see Fig. 3). The domain of this theory is the set of non-empty regular closed sub-

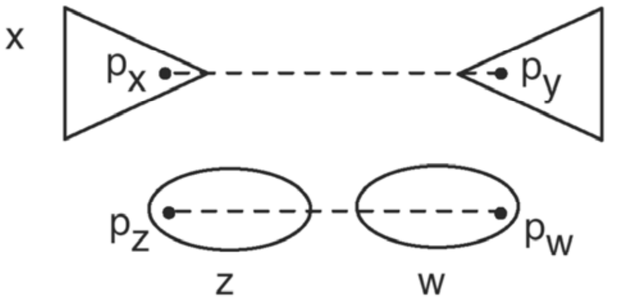


Figure 3: x, y *and* z, w *are conjugate, Conj(x, y, z, w,*

---

[6] Formally, *S(x)* is translated as: there exists a point $c \in \mathfrak{R}^n$ and a value $r \in \mathfrak{R}^+$ (the positive reals) such that *x = ball(c, r))* (see Fig. 1).

[7] Recall that *x* is an extended region and should not be confused with a set of points. However, the formal interpretation takes *x* to be a set of points and so the informal reading is justified.

sets of $\Re^n$ that are self-connected. Compared to T1, the theory discharges scattered regions.

T4: Next, we have the mereogeometry introduced in [De Laguna 1922] and further developed in [Donnelly 2001]. This time we have just the primitive dubbed **can-connect** and indicated by *CCon*. *CCon(x, y, z)* stands for "region *x* can connect both regions *y* and *z*". The idea is that the length of the diameter of region *x* is at least as the distance between regions *y* and *z*: intuitively, if this holds, one can 'move' *x* in a position where it is in contact with both *y* and *z* (see Fig. 4). The domain is more restricted than those seen so fare: it takes only non-empty regular closed subsets of $\Re^n$ that are both self-connected and finite. Compared to T1, the theory considers closed regions only and discharges both scattered regions and infinite regions.

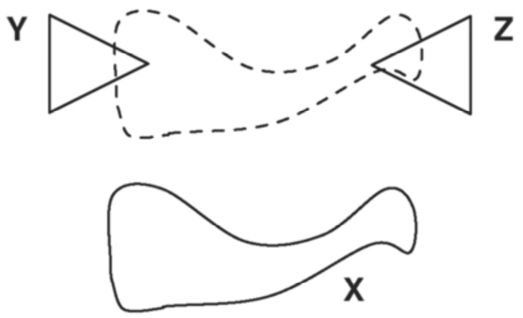


Figure 4: x *can connect* y *and* z, CCon(x, y, z)

T5: The system introduced in [Van Benthem 1983] was later further developed in [Aurnague *et al.* 1997]. In this mereology, the primitives are the binary relation *C* of **connection** and the ternary relation *Closer* of **closerness**. *C(x, y)* stands for "region *x* is connected to region *y*" and *Closer(x, y, z)* for "region *x* is closer to region *y* than to region *z*" (see Fig. 5). The domain for this theory is the set of non-empty regular subsets of $\Re^n$. That is, it is larger that the domain of T1 since the latter takes the open regular regions only.

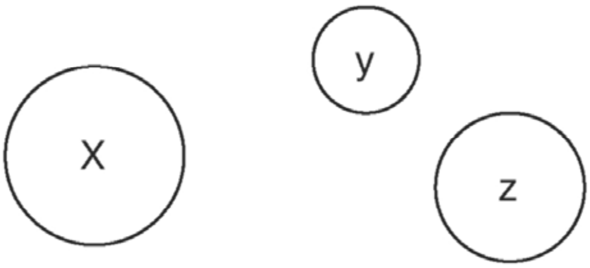


Figure 5: x *is closer to* y *than to* z, Closer(x, y, z)

T6: Finally, we consider the system given in [Cohn 1995, Cohn *et al.* 1997a & 1997b]. Here there are two primitives: the binary relation *C* of **connection** already seen in T5 and the binary relation *ConvH* of **convex-hull**: ConvH(x, y) stands for "region *x* is the convex hull of region *y*" (see Fig. 6). The theory takes as domain the set of non-empty regular open subsets of $\Re^n$ like in T1.

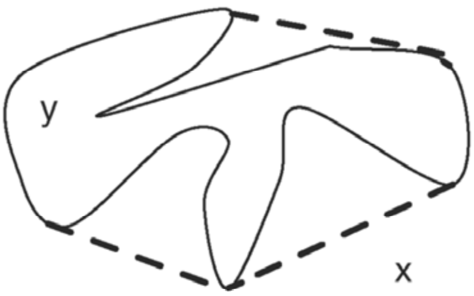


Figure 6: x *(corresponding to region* y *plus the areas enclosed by the dash lines) is the convex hull of* y*,* ConvH(x, y)

Although these theories adopt quite disparate primitives, in [Borgo & Masolo, *to appear*] it has been shown that they are closely tied. To appreciate their interrelationships, we need first to introduce some notions.

Informally, the comparison between two systems consists in showing that everything that can be said in one system can be said in the other, and furthermore, that everything which holds in one system holds in the other as well. In logic, this result is usually obtained via the notion of ‘explicit definition’ that amounts to showing that what is primitive in one system can be defined in the other. However, here the comparison is complicated by the lack of axiomatization of some theories, a difficulty increased by the fact that the theories rely on different domains of discourse. For this reason, we generalize the notion of explicit definition as follows.

Definition 3.1 *If A is a primitive of a theory T, we say that A is explicitly definable in another theory T’ for a domain D if there exists an expression Φ in the language of T’ such that the interpretations of A and Φ are equivalent for their structures with domain D.*

If we forget the reference to domains, the notion of explicit definition leads to the classical notion of equivalence among theories. Two theories are equivalent if all the primitives of the first are explicitly definable in the second, making the first a subtheory of the latter, and *vice versa*. However, for mereogeometries the dependence on the domain is crucial. Then, we need to introduce the generalized notions of subtheory and equivalence.

Definition 3.2 *A theory T is a subtheory of T’ for domain D if every primitive T has an explicit definition in T’ for that domain.*

Now, it becomes possible to capture a notion of equivalence, called conceptual equivalence, that is suited to mereogeometries. Basically, it relativizes equivalence to logical structures.

Definition 3.3 *Let T and T’ be theories with domains D_i and D_j , respectively, T and T’ are conceptually equivalent if T is a subtheory of T’ and T’ is a subtheory of T with respect to both D_i and D_j.*

The results in [Borgo & Masolo, *to appear*] can now be formulated as follows.

Theorem 3.1

- Theories T1, T2, T3, T4, T5 are equivalent;
- T6 is a subtheory of all theories T1, T2, T3, T4, T5;
- T1, T2, T3, T4 are conceptually equivalent.

Theories T1, T2, T3, T4 have the characteristics of a complete geometry and are called full mereogeometries. The other two systems are set apart for different reasons. T5 is as strong as the previous but it is associated to a much richer domain where regions of different dimensions can coexist. The analysis of this domain seems to require further considerations. Theory T6 is representative of a large number of mereogeometries. This and many other systems in the literature, e.g., the Lines Of Sight [Galton 1994] and ROC [Randell & Witkowski 2006], have limited expressiveness, and their position in the landscape of mereology is not yet clear. Nonetheless, subsystems of full mereogeometries are of great interest since they capture a particular perspective on space whose motivations come from very active areas of research like robotics and qualitative aspects of human perception.

4 Topics across Mereogeometry and Pictorial Morphology

We have seen that mereogeometry stems from the need to study space independently from entities and notions that are out of reach for human perception. This research has two major motivations, which we have not set apart yet, namely the study of space from a cognitive perspective, and the representation of space in qualitative fashion.

In the *cognitive perspective*, which has implicitly driven the exposition of section 2, the goal is to find a formal characterization of space, as experienced by humans, which rely upon entities and relations that are as much as possible under human perception and (direct) cognitive grasp. The fact that there are several alternative options (as seen in section 3) does not weaken the goal since this area did not suffered from the myth of the ‘true model of space’, a myth that affected the whole history of Euclidean geometry. The other approach is dubbed *qualitative*. Qualitative systems are formalisms widely studied in the artificial intelligence community since they embrace a perspective strongly focused on the balance between expressiveness and (effective) computability. In this case, one looks at formal representations of space where, roughly speaking, one can represent a limited set of geometrical properties (like those relevant to perception, navigation or conceptualization of external reality), without the formal complexity intrinsic to point-based geometry. In a nut, the goal is to find ways to represent limited amounts of spatial information avoiding computationally expensive languages.

Both these views have import in the area of pictorial morphology as is pointed out in [Schirra 2005]. On the one hand, the search for grounding *pixemes* (either as primitives or as prototypical) naturally leads to a discussion that matches the debate on basic geometrical entities. On the other hand, the need of rendering and understanding complex images in a computational setting suggests (at least in theory) the existence of a limited number of basic pixemes that can be combined via a formal calculus of limited complexity, and thus, hopefully, being qualitative. Of course, there is much more in pictorial morphology than this as it was clear from the beginning, see for instance the seminal work of Goodman [1968]. While mereogeometry stops at the geometrical aspects of physical objects and their relationships, pictorial morphology has to take into account other elements like granularity (which may affect very basic properties as connectedness among entities, i.e., the topology itself) and appearance (from which the difference between resemblance in geometry and in perception). Indeed, mereogeometry inherits from standard geometry the primary interest in *loci* and *shape description* including features like, e.g., linear borders (the betweenness relationship has been studied both in point based and in region-based geometry). The goal, from this perspective, is a formalization of the necessary and sufficient conditions for classifying relevant entities and entity dispositions. Pictorial morphology comes from a broader view where the issue of entity description is subordinated to the primary goal of entity *recognition*. The interaction between an image and what it depicts is highly intertwined with non-geometrical aspects like experience, expectations, and commonsense reasoning; aspects that go beyond even visual perception. In this context, one cannot disregard forces like gravity (which is intrinsic in the representation and perception of supported and supporting objects), properties like orientation (which is archetypal in animals and buildings), emerging features like density (related to the distribution of objects) and other perceptually relevant characteristics like colors.

Limiting our discussion on the common elements in mereogeometry and pictorial morphology, we should ask what can be learned from the latest results in the study of space. Perhaps, the first observation to make is that mereogeometry corroborates a conclusion that has puzzled researchers in pictorial morphology: the lack of constraints on the choice of primitives. In these domains one arrives easily to equivalent formalisms starting from quite disparate assumptions. It follows that the choice of primitives cannot rely on purely formal properties, it must be supported by arguments and observations from other perspectives like those embedded into the cognitive, evolutionary, mental, and perceptive views. In geometry and mereogeometry we have observed the development of several geometrical systems which, exploiting disparate primitives, naturally lead to formal geometries of equivalent expressiveness.[8] These primitives may capture comprehensive shape descriptions like ‘being a sphere’ [46], may concentrate on local features like ‘having cavities’ [Cohn 1995], ‘having a corner’ [Eschenbach *et al.* 1998] (‘forming a right angle’ in Euclidean geometry [Scott 1997]), or on global properties like ‘being fully symmetrical’ and its opposite,’ being totally asymmetrical’. The expression ‘fully symmetrical’ is here used to indicate a region that is symmetrical with respect to a given point as well as with respect to all the lines (planes and hyper-spaces in general) through that point. Needless to say that the notion of ‘fully symmetrical region’ brings us quickly to the definition of sphere (in any dimension). The expression ‘totally asymmetrical’ instead is satisfied by a region that is asymmetrical with respect to any point, line, plane and hyper-space in general.

---

[8] In the light of section 3, equivalence is to be intended “modulo” the choice of the domain.

All these approaches have been analyzed in two directions: formal expressiveness, and cognitive role. The first issue has brought a series of scattered results that are slowly building up the landscape of the mereogeometries. Cognitively, the results are less promising: no cognitive system seems to be identified as central or primary. One can concentrate on direction or orientation, on distance or size or shape; take into account vagueness, disposition, forms of resemblance etc. The result will be a system, perhaps unusual, perhaps hard to compare to well known geometries and yet it will have that flavor of cognitive adequacy or conformity that will prevent us from discharging it.

5 Mereogeometry as a Tool for Pictorial Morphology

We conclude this excursus on mereology and its relationship to pictorial morphology with a few observations that suggest how results in the first area can help casting light into the second. Although the discussion can apply to the variety of perspectives embedded in pictorial morphology, let us focus on a concept like *feature*, and on the distinction between *content-bearing features* and *noncontent-bearing features*. It is clear that the analysis of resemblance and systems of pictorial representation must make clear what types of features there are and what information they carry. Also, from the arguments presented in this paper, we should not expect mereogeometry to answer the main questions. Still, we know that we can positively look at mereogeometry for important hints. For instance, the settling of the structure and properties of projective mereogeometry (a subdomain that is still not well understood) will help us in isolating the spatial features of prospective representations and how these group together by forming interconnected systems. Such a work is needed to clear up the types of resemblance that properly depend on these features and the interrelations within these types. Similarly, mereology will not tell us what images are or what their content is. Nonetheless, it can be an important tool to determine why pictorial representations follow some spatial rules but not others, and why some relations are necessarily intrinsic (think of the relationship among a picture and its parts, their relative sizes and topological properties) while others may not be.

References

Aurnague, M. & Vieu, L. & Borillo, A.: La représentation formelle des concepts spatiaux dans la langue. In M. Denis (ed.): *Langage et Cognition Spatiale*. Paris, 1997, 69–102.

Bennett, B.: A categorical axiomatisation of region-based geometry. *Fundamenta Informaticae*, 46:145–158, 2001.

Bennett, B. & Cohn, A. G. & Torrini, P. & Hazarika, S. M.: Describing rigid body motions in a qualitative theory of spatial regions. In Kautz, H. A. & Porter, B. (eds.): *National Conf. on AI (AAAI'00)*. AAAI Press / MIT Press, 2000, 503–509.

Bennett, B. & Cohn, A. G. & Torrini, P. & Hazarika, S. M.: A foundation for region-based qualitative geometry. In Horn, W. (ed.): *European Conf. on AI (ECAI'00)*. IOS press, 2000, 204–208.

Borgo, S. & Guarino, N. & Masolo, C.: A pointless theory of space based on strong connection and congruence. In Carlucci Aiello, L. & Doyle, J. & Shapiro, S. C. (eds.): *Intern. Conf. on Principles of Knowledge Representation and Reasoning (KR'96)*. Morgan Kaufmann, 1996, 220–229.

Borgo, S. & Masolo, C.: Full mereogeometries. *Journal of Philosophical Logic* (to appear).

Casati, R. & Varzi, A.: *Parts and Places. The Structure of Spatial Representation*. MIT Press, Cambridge, MA, 1999.

Clarke, B. L.: A calculus of individuals based on connection. *Notre Dame Journal of Formal Logic*, 22:204–218, 1981.

Clarke, B. L.: Individuals and points. *Notre Dame Journal of Formal Logic*, 26:61–75, 1985.

Cohn, A. & Bennett, B. & Gooday, J. & Gotts, N.: Representing and reasoning with qualitative spatial relations. In Stock, O. (ed.): *Spatial and Temporal Reasoning*. Kluwer, Dordrecht, 1997, 97–134.

Cohn, A. G.: Qualitative shape representation using connection and convex hulls. In Amsili, P. & Borillo, M. & Vieu, L. (eds.): *Time, Space and Movement: Meaning and Knowledge in the Sensible World*, Part C. IRIT, 1995, 3–16.

Cohn, A. G.: Formalising bio-spatial knowledge. In Smith, W. C. & Smith, B. (eds.): *Intern. Conf. on Formal Ontology in Information Systems (FOIS'01)*. ACM Press, 2001, 198–209.

Cohn, A. G. & Bennett, B. & Gooday, J. M. & Gotts, N.: Rcc: a calculus for region based qualitative spatial reasoning. *GeoInformatica*, 1:275–316, 1997.

Cohn, A. G. & Hazarika, S. M.: Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46(1-29), 2001.

Cristani, M. & Cohn, A. G. & Bennett, B.: Spatial locations via morpho-mereology. In Cohn, A. G. & Giunchiglia, F. & Selman, B. (eds.): *Intern. Conf. on Principles of Knowledge Representation & Reasoning (KR'00)*. Morgan Kaufmann, 2000, 15–25.

De Laguna, T.: Point, line, and surface. a sets of solids. *The Journal of Philosophy*, 19:449–461, 1922.

Donnelly, M.: An axiomatic theory of common-sense geometry. PhD Thesis, 2001.

Donnelly, M.: On parts and holes: The spatial structure of the human body. In Fieschi, M. & Coiera, E. & Li, Y.-C. J. (eds.): *MedInfo 04*. IOS Press, 2004.

Dugat, V. & Gambarotto, P. & Larvor, Y.: Qualitative theory of shape and orientation. In Rodriguez, R. V. (ed.): *Workshop on Hot Topics in Spatial and Temporal Reasoning (IJCAI'99)*, 1999, 45–53.

Eschenbach, C. & Habel, C. & Kulik, L. & Leßmöllmann, A.: Shape nouns and shape concepts: A geometry for 'corner'. In Freksa, C. & Habel, C. & Wender, K. (eds.): *Spatial Cognition*. Springer, Berlin, 1998, 177—201.

Galton, A.: Lines of sight. In *AISB Workshop on Spatial & Spatio-Temporal Reasoning*, 1994.

Galton, A.: *Qualitative Spatial Change*. Oxford University Press, New York, 2000.

Galton, A.: Multidimensional mereotopology. In D. D., W. C., & W. M.A. (eds.): *International Conference on the Principles of Knowledge Representation and Reasoning (KR'04)*. AAAI Press, 2004, 45–54.

Goodman, N.: *Languages of Art: An Approach to a Theory of Symbols*. Hackett, Indianapolis, 1968.

Gotts, N.: Formalizing commonsense topology: The inch calculus. In Kautz, H. & Selman, B. (eds.): *International Symposium on Artificial Intelligence and Mathematics (AI/MATH'96)*, 1996, 72–75.

Grzegorczyk, A.: Axiomatizability of geometry without points. *Synthese*, 12:228–235, 1960.

Kant, I.: *Critique of Pure Reason*. (Engl.: W. S. Pluhar), Hackett, Indianapolis, 1996 (1787).

Li, S. & Ying, M.: Region connection calculus: Its models and composition table. *Artificial Intelligence*, 145:121–146, 2003.

Lobachevskij, N. I.: *New principles of geometry with complete theory of parallels*, volume 2. Polnoe sobranie socinenij, 1835.

Muller, P.: Eléments d'une théorie du mouvement pour la modélisation du raisonnement spatiotemporel de sens commun. PhD thesis, 1998.

Muller, P.: A qualitative theory of motion based on spatio-temporal primitives. In Cohn, A. G. & Schubert, L. & Shapiro, S. C. (eds.): *International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 131–141. Morgan Kaufmann, 1998.

Nicod, J.: *La Géométrie dans le monde sensible*. Presse Universitaire de France, Paris, 1924.

Pratt-Hartmann, I. & Lemon, O.: Ontologies for plane, polygonal mereotopology. *Notre Dame Journal of Formal Logic*, 38:225–245, 1997.

Pratt-Hartmann, I. & Schoop, D.: Expressivity in polygonal, plane mereotopology. *Journal of Symbolic Logic*, 65:822–838, 2000.

Randell, D. A. & Cohn, A. G.: Modelling topological and metrical properties in physical processes. In Brachman, R. J. & Levesque, H. J. & Reiter, R. (eds.): *International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*. Morgan Kaufmann, 1989, 357–368.

Randell, D. A. & Cohn, A. G.: A spatial logic based on regions and connections. In Nebel, B. & Rich, C. & Swartout, W. (eds.): *Internat. Conf. on Principles of Knowledge representation and Reasoning (KR'92)*. Morgan Kaufmann, 1992, 165–176.

Randell, D. A. & Witkowski, M.: Using occlusion calculi to interpret digital images. In *ECAI'06*, 2006, 432–436.

Renz, J. & Rauh, R. & Knauff, M.: Towards cognitive adequacy of topological spatial relations. In Freksa, C. & Brauer, W. & Habel, C. & Wender, K. (eds.): *Spatial Cognition II*. Springer, 2000, 184–197.

Schirra, J.R.J.: *Foundation of Computational Visualistics*. DUV, Wiesbaden, 2005.

Schlick, M.: *Naturphilosophie*. Engl. Transl. in *Philosophical Papers II*, 1925-1936", ed. by Mulder, H. L. & Van de Velde-Schlick, B., Reidel, Dordrecht, 1925.

Schmidt, H.: *Axiomatic Characterization of Physical Geometry*. Springer, Berlin, 1979.

Schulz, S. & Hahn, U.: Mereotopological reasoning about parts and (w)holes in bioontologies. In Smith, W. C. & Smith, B. (eds.): *International Conference on Formal Ontology in Information Systems (FOIS'01)*. ACM Press, 2001, 210–221.

Scott, D.: A symmetric primitive of euclidean geometry. *Indagationes Mathematicae*, 18:456-461, 1956.

Simons, P.: *Parts: a Study in Ontology*. Clarendon Press, Oxford, 1987.

Smith, B.: Basic concepts of formal ontology. In Guarino, N. (ed.): *Proceedings of the First International Conference FOIS 1998*. IOS Press, 1998, 19–28.

Smith, B. & Varzi, A.: The niche. *Nous*, 33:214–238, 1999.

Stock, O. (ed.): *Spatial and Temporal Reasoning*. Kluwer, Dordrecht, 1997.

Tarski, A.: Foundations of the geometry of solids. In J. Corcoran, editor, *Logic, Semantics, Metamathematics*. Oxford University Press, Oxford, 1956, 24–30.

J. Van Benthem. *The Logic of Time*. Kluwer, Dordrecht, 1983.

L. Vieu. Spatial representation and reasoning in artificial intelligence. In Stock, O. (ed.): *Spatial and Temporal Reasoning*. Kluwer, Dordrecht, 1997, 5–41.

Whitehead, A. N.: *Process and Reality. An Essay in Cosmology*. Macmillan, 1929.

[Table of Content]

Winfried Kurth

Specification of Morphological Models with L-Systems and Relational Growth Grammars

Abstract

Among the techniques for the creation of photorealistic virtual organisms, particularly plants, and in scientific models of vegetation structure, rule-based specifications (formal grammars) play a prominent role. Lindenmayer systems (L-systems) are the most widespread formalism of this sort, but certain types of graph grammars, combined with standard object-oriented programming, offer even more possibilities to specify rule-driven developments of 3-dimensional arrangements, morphology of virtual organisms and underlying processes like, e.g., metabolic reactions. Examples of grammar rules and the virtual geometrical structures generated from them, all realizable with the open-source software GroIMP (www.grogra.de), are shown. This grammar-based approach is often not immediately used for the direct specification of a picture as a pattern of graphical elements in a plane, but for virtual 3-D scenes, which are then rendered visible using standard techniques of geometry-based computer graphics.

1 Introduction: Rule-based modelling of development

The programme for a computer-based simulation of a process is often specified by writing down the elementary steps of calculation in a prescribed order, which is to be applied when the machine executes them. This order can include the use of conditional branching and loops. Furthermore, in this classical programming style, commands have usually the meaning that the state of the machine – manifested, e.g., in the values of some memory cells – is changed in a predefined manner. This programming paradigm is called “imperative” or “von Neumann programming”, and can be very useful in technical calculations or for simulations in physics.

However, when living organisms and the development of their morphological structure are to be modelled, another sort of programming seems to be more natural. Let us consider, for instance, a growing tree: All parts of the organism coexist, and the young shoots of the tree grow all in parallel, often according to the same pattern. An intuitive way to specify this behaviour is to list a number of rules for growth of single buds and shoots (or whatever organs are considered as the basic constituents), and to let the computer apply them in parallel to all tree organs, wherever they are applicable. When the growth flush of the next year is to be simulated, the application of these rules is to be iterated. Here, the order in which the rules of growth are written down is not important: The computer is expected to pick those rules which are applicable in a given situation, and to use them regardless of their position in a list. This "rule-based" programming paradigm is well known in other branches of information science: *Grammars* of natural languages and of programming languages are used in a similar manner, with the aim to deduce all correctly formed sentences. Another example is the programming language PROLOG, where logical rules are applied to generate automatic proofs of statements. In all these cases, some structure – a botanical tree / a sentence / a logical formula – is transformed or *rewritten* by the application of rules. The systems of rules, or grammars, are therefore also called "rewriting systems". Rule-based programming can be a more intuitive way to specify models of natural phenomena, because we do not need to bother about a specific order of execution of commands. The rules work at a higher level of abstraction.

The biologist Aristid Lindenmayer invented in 1968 a special sort of grammar, later called L-system, to describe the growth of arrangements of plant cells [Lindenmayer 1968]. At that time, the notion of formal grammar, developed by Noam Chomsky for natural languages, was already known. However, in a Chomsky grammar, normally only one rule is applied in each deduction step. In contrast, L-systems work in a parallel manner, thus reflecting the parallelism of growth in plants: That means, in every time step all constituents of the virtual plant where some rule is applicable are transformed according to that rule. (If there are some objects on which no rule can be applied, it is assumed that these objects are just resting: They remain unchanged.)

Later on, Lindenmayer's formalism, which is basically a string-rewriting mechanism, was extended.[1] A command language for a geometrical interpretation of strings was introduced to give a precise definition of the morphological meaning of the structures obtained from L-system application. We will briefly introduce this "Turtle Geometry" in Chapter 2. In Chapter 3, L-systems will be exactly defined, and we will see some simple examples. Several extensions of the original concept were used to solve various problems in the modelling of plant growth and architecture; some of these extensions will be explained and demonstrated in Chapter 4. An important generalization, which is currently still in the focus of research, is introduced in Chapter 5: "Relational Growth Grammars" (RGG), a variant of graph rewriting systems. These grammars overcome some of the limitations of L-systems and can be used to connect different levels of the organization of plants in a unifying

---

[1] see [Prusinkiewicz & Lindenmayer 1990] for references and historical remarks.

model framework: Genetic processes influencing metabolism, metabolic reaction networks influencing macroscopic growth and morphogenesis. Simulation models based on this sort of grammar representation can not only produce even more realistic images of plants and plant communities, but will also aid the biologists in checking hypotheses and designing new experiments. A discussion of possible future trends in modelling morphological phenomena and of the relation of the rule-based programming paradigm to picture morphology will close the article.

2 Turtle geometry

To establish a connection between the language of *character strings* and the language of *geometrical forms*, a simple alphabet of commands, each with a geometrical meaning, is defined. Using these commands, we build programmes in a strictly imperative manner, which are interpreted by a virtual drawing device, called the "turtle" [Abelson & diSessa 1982]. The turtle is equipped with a simple memory containing information about the length *s* of the next line to be drawn, its thickness *d*, its colour *c*, the turtle's current position on the plane, its current direction of moving, etc. Among the possible commands are:

`M0`	move forward by length *s* (without drawing)
`F0`	move forward and draw simultaneously a line of length *s*
`M(`*a*`)`	move forward by length *a* (without drawing); the explicitly specified number *a* overrides the turtle's inherent *s*
`F(`*a*`)`	move forward and draw simultaneously a line of length *a*
`L(`*a*`)`	overwrite *s* by the value *a*
`D(`*a*`)`	overwrite *d* by the value *a*
`P(`*a*`)`	overwrite *c* by the value *a* (interpreted as a colour index)
`RU(`*a*`)`	rotate clockwise by the angle *a* (around the "up" axis, which is perpendicular to the plane where the turtle is moving)
`Sphere(`*a*`)`	produce a filled circle with radius *a* around the current position without moving

The zero in **`M0`** and **`F0`** means that there is no explicit argument; instead, the memorized "state variable" *s* of the turtle is used. Strings composed of these commands can be used to specify structures made of consecutive lines with changing length, thickness, and visibility. Each such string describes a *static* geometrical structure. E.g., the string

```
L(100) D(3) RU(-90) F(50) RU(90) M0
RU(90) D(10) F0 F0 D(3) RU(90) F0 F0
RU(90) F(150) RU(90) F(140) RU(90) M(30)
F(30) M(30) F(30) RU(120) M0 Sphere(15)
```

describes the structure in Figure 1.

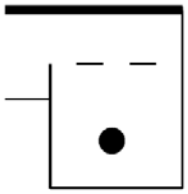


Figure 1: The result of a simple turtle command sequence (see text)

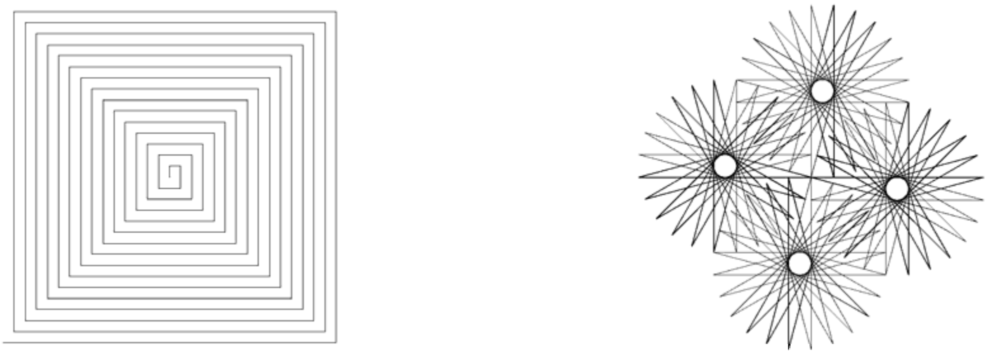


Figure 2: (a) A spiral specified by a simple iterative turtle programme, (b) the result of another iterative turtle programme (after [Goel & Rozehnal 1991])

As in other imperative programming languages, *loops* can be used to abbreviate iterated parts of the string: **`for (i:(1:`***n***`))(`***X***` )`** generates *n* replications of the string *X*. Hence, the turtle command programme

```
L(100) for (i:(1:30))
         ( for (j:(1:i)) (F0) RU(90)
           for (j:(1:i)) (F0) RU(90) )
```

generates the spiral in Figure 2a, and

```
L(100) for ((1:20))
         ( for ((1:36)) ( F0 RU(165) F0 RU(165) ) RU(270) )
```

generates the pattern in Figure 2b.

To overcome the restriction to strictly linear forms, the possibility of branching is introduced by the special turtle commands “[“ and “]”: When the turtle encounters “[“, its current state (including the values of *s*, *d*, *c* etc.) is stored on a stack. The following string can be seen as a branch which ends when “]” is encountered: Then the stored state is taken from the stack and replaces the turtle state which was obtained during the drawing of the branch. This means that the turtle “jumps back” to its old position and resumes its operation as if the construction of the branch since “[“ would not have taken place. Figure 3 shows the turtle interpretation of the following string:

```
F(50) [ RU(60) P(4) F(20) ] RU(-30) F(50) .
```

After the vertical segment of length 50, the smaller, red branch to the right (coloured according to the command **`P(4)`**) is constructed. After the closed bracket, the turtle resumes its old position and follows the commands **`RU(-30) F(50)`** to draw the upper-left part of the structure.

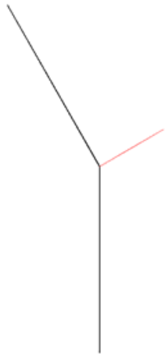


Figure 3: A branched structure (see text)

The turtle can also be guided to draw structures in three dimensions. For this purpose, two further rotation commands are introduced: **`RL(`***a***`)`** and **`RH(`***a***`)`**, which rotate the turtle around an axis pointing (initially) to the left, respectively around its current head direction.[2]

3 L-Systems

Lindenmayer systems (L-systems) are *parallel rewriting systems on strings*. Mathematically, a "pure" L-system (without geometrical interpretation) consists of 3 components: an alphabet Σ which contains the basic symbols that are to be used to build strings, a start string called "Axiom", and a finite set of rules, each of which having the form

symbol ==> string of symbols;

and the symbols are taken from Σ here. In a deterministic L-system, the symbol on the left-hand side (l.h.s.) of each rule must be different from those of all other rules. An *application step* of the L-system to a given string *s* consists of the simultaneous replacement of all symbols in *s* occurring as a l.h.s. of a rule by their corresponding right-hand side (r.h.s.), whereas symbols which cannot be replaced with the help of a rule remain unchanged. By starting with the start string of the L-system and iteratively performing one application step to the result of the preceding one, we obtain the *developmental sequencea* of strings generated by an L-system:

$\text{Axiom} \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$

For example, let us consider the L-system with the alphabet Σ = { **`A`**; **`B`** }, *Axiom* = **`A`**, and with the two rules

`A ==> B`

`B ==> AB.`

The resulting developmental sequence is

---

[2] See the tutorial included in the GroIMP software, freely available under www.grogra.de, for further details about turtle commands.

`A → B → AB → BAB → ABBAB → BABABBAB → ...`

Following Lindenmayer's original intentions, **`A`** and **`B`** can be interpreted as two different cell types of filamentous organisms (e.g., algae). The rules say that a cell of type **`A`** can grow into a cell of type **`B`**, and a type **`B`** cell can divide into two cells of type **`A`** and **`B`**, respectively. The developmental sequence then reflects the growth of the filament of cells in discrete time steps. (Note that the number of cells generated in this sequence grows according to the Fibonacci sequence: 1, 1, 2, 3, 5, 8, 13, ..., where each number is the sum of its two predecessors.)

To produce more interesting structures from L-systems than just linear filaments of cells, Alvy Ray Smith [Smith 1984] and later Prusinkiewicz and Lindenmayer [1990] added turtle geometry as a fourth component to Σ, *Axiom* and the rule set. Turtle geometry serves as a geometrical interpretation, i.e., as a means to associate with each string (particularly with each s_i from the developmental sequence above) a geometrical structure S_i in 2- or 3-dimensional space. This is accomplished by letting the alphabet Σ contain the set *T* of all turtle commands. The turtle then separately interprets the strings s_i obtained from the L-system, i.e., they are scanned from left to right, and the geometrical structure S_i is constructed by following the occurring commands. Symbols from Σ that are not in T are simply ignored by the turtle. Hence we have the following scheme of interpreted L-system application:

Here, the dotted green arrows stand for the turtle interpretation process.

The first example (after [Prusinkiewicz & Hanan 1989, p. 25]) will demonstrate this mechanism: Let the rules of our L-system be

```
Axiom ==> L(100) F0 and
F0 ==> F0 [ RU(25.7) F0 ] F0 [ RU(-25.7) F0 ] F0 .
```

Figure 4 shows the resulting structures S_1, S_2, S_3 and S_4.

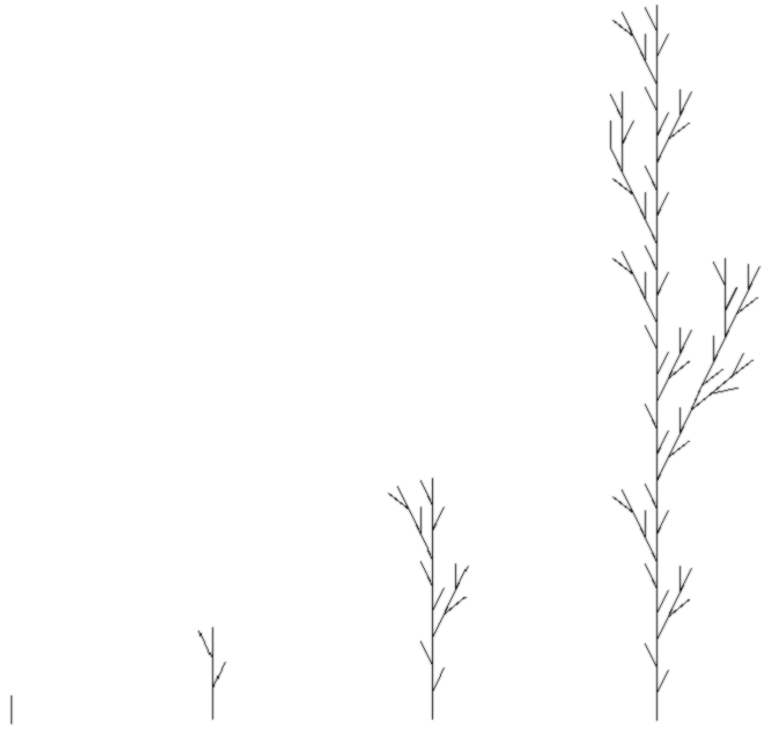


Figure 4: A developmental sequence of branching structures in the plane, generated by a simple L-system (see text)

The next two examples use L-systems to generate plane-filling curves. Both make use of the possibility, given in the programming language XL [Kniemeyer 2007], to let symbols (in this context called "modules") inherit properties from other symbols. Such an inheritance from `A` to `B` is expressed in the form

```
module B extends A;
```

and this is a formalism typical for object-oriented programming. Its purpose in the following examples is simply the abbreviation of commands.

A so-called *hexagonal Gosper curve* is derived from

```
module A extends F0;
module B extends F0;
module C extends RU(60);
module D extends RU(-60);
Axiom ==> L(100) A;
A ==> A C B C C B D A D D A A D B C;
B ==> D A C B B C C B C A D D A D B;
```

with the result after 4 steps shown in Figure 5a (after [Prusinkiewicz & Hanan 1989, p. 19]), and the second curve resembles a traditional Indian *kolam* pattern (see Figure 5b and [Ascher 2003]), called "Anklets of Krishna" (after [Prusinkiewicz & Hanan 1989, p. 73]), and is derived from


a *b*

Figure 5: Two plane-filling curves obtained from L-systems, see text

```
module R extends RU(-45);
module A extends F(10);
Axiom ==> L(100) R X R A R X;
X ==> X F0 X R A R X F0 X;  .
```

4 Extensions of the L-system concept

4.1 Stochastic L-systems

Geometrical structures produced by the simple forms of L-systems that we have presented so far show a high degree of regularity. In real-world patterns, however, we have often variability and "noise", producing deviations from strict regularity. A first attempt to reflect this "noise" in a model is the inclusion of randomness. The computer can generate pseudo-random numbers, appearing as if they do not follow any predictable pattern, and this form of irregularity can be introduced in rewriting systems – either by directly using pseudo-random numbers as parameters (e.g., of L or RU commands) or by making rule application depend on some "oracle" driven by pseudo-random numbers. For example, let us consider the deterministic L-system

```
float c = 0.7;
Axiom ==> L(100) D(5) A;
A ==> F0 LMul(c) DMul(c) [ RU(50) A ] [ RU(-10) A ].
```

(Here, "`float`" declares a floating-point variable `c` which gets the value 0.7 and is used in the second rule; "`LMul(c)`" multiplies the current length *s* of the turtle steps with this number, and "`DMul(c)`" analogously for current thickness *d*.) The tree-like structure produced by this L-system looks very regular (Fig. 6a).

If we exchange the second rule by

```
A ==> F0 LMul(c) DMul(c)
      if (probabiliy(0.5)) ( [ RU(50) A ] [ RU(-10) A ] )
      else ( [ RU(-50) A ] [ RU(10) A ] );
```

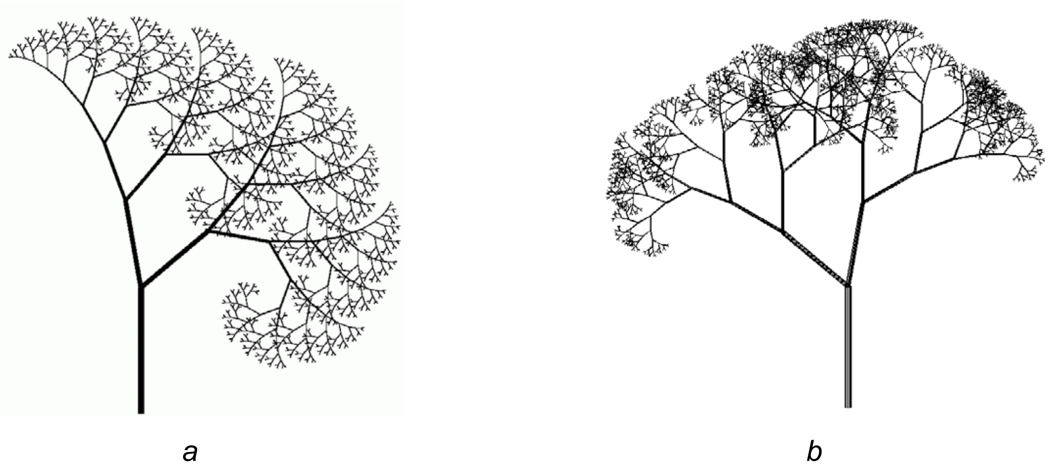


Figure 6: Tree-like structures generated from an L-system (a) deterministic version, (b) stochastic version

the orientation of the two branches, specified by the "`RU`" commands, is switched (or not) in an arbitrary manner in each new bifurcation of the tree. Each of the two orientations is chosen with equal probability 0.5, as if the outcome would depend on coin-tossing, and the resulting structure has already a somewhat more natural look (Fig. 6b).

Of course, it would be possible to increase the irregularity even further, e.g., by replacing the constant c above by "`random(0.3, 1)`", a function call which gives back pseudo-random numbers with uniform distribution between 0.3 and 1. Using the same formalism, it is also easily possible to simulate random walks in the plane or in space (e.g., Brownian motion in physics), or to generate more-or-less-controlled random distributions of small objects in an area – what is called a "point process" in geostatistics.

A very simple example is given by the following L-system, consisting of only one rule:

```
Axiom ==> D(0.5) for ((1:300))
          ( [ Translate(random(0, 100), random(0, 100), 0)
              F(random(5, 30)) ] );
```

that generates 300 vertical lines with random lengths between 5 and 30 units at random positions on a 100 x 100 square field (Fig. 7). Here, the command "Translate" works like "M", but the direction of the translation is given in absolute coordinates (*x*, *y*, *z*), not as a multiple of the current turtle head vector.

4.2 Parametric L-systems

We have already used parameters with numerical values in turtle commands like **`L`**, **`LMul`**, **`D`** or **`F`** in the examples above. If we permit the use of such parameters in connection with other symbols, too, the capacity of our rewriting mechanism to perform calculations of all kinds is greatly enhanced. For example, in the next L-system, which produces a fractal

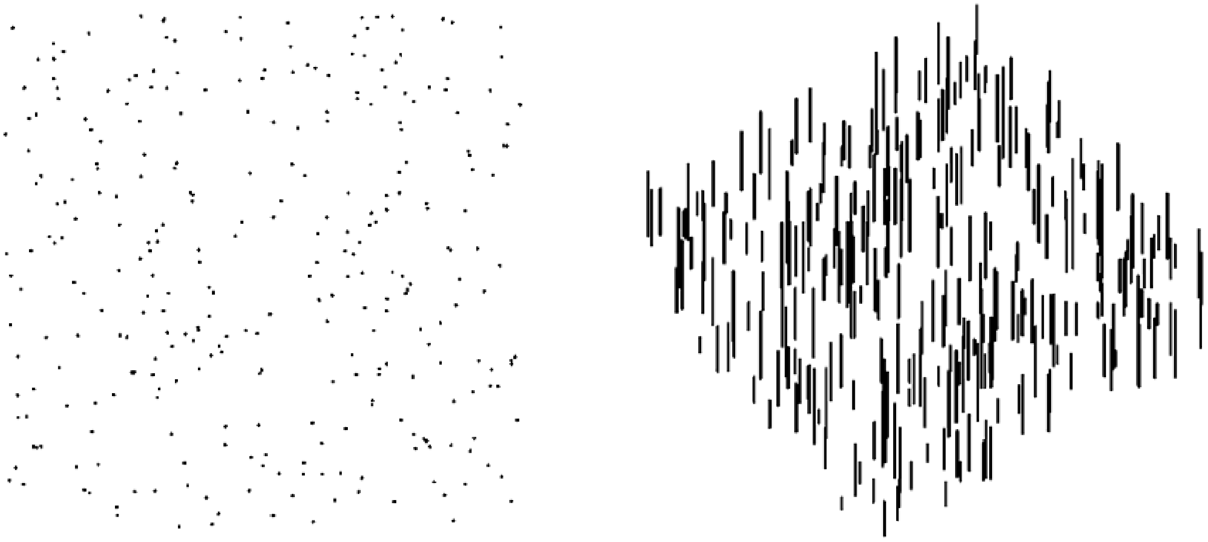

a b

Figure 7: A random pattern of vertical lines on a quadratic area
(a) view from above, (b) slanted view

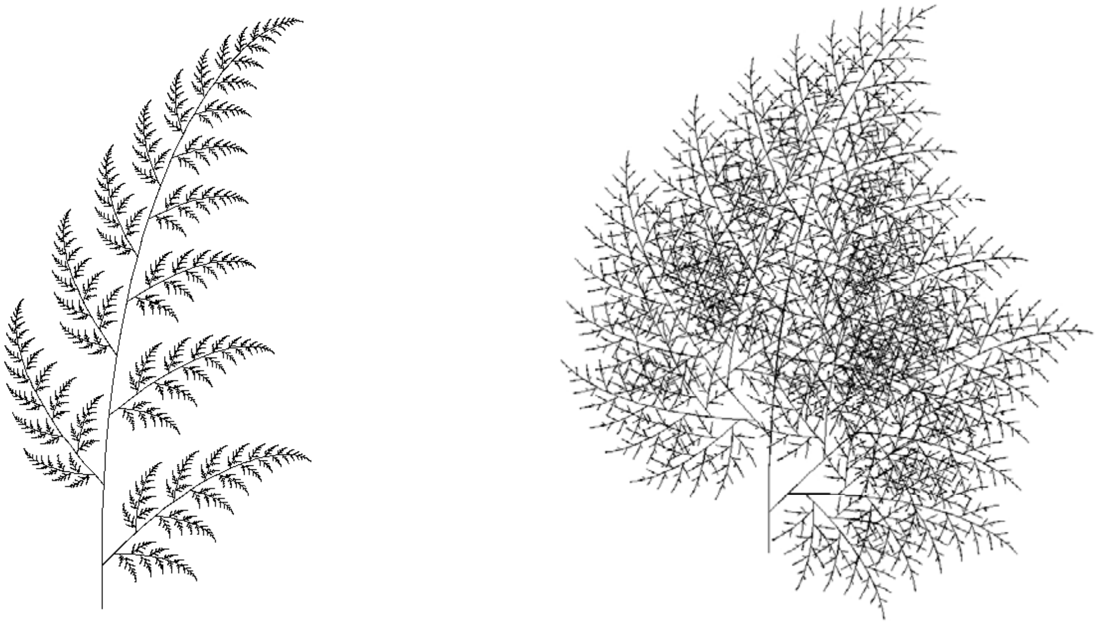
structure resembling a fern leaf (Fig. 8a), we use two integer parameters *t* and *k* for the symbol **A**. The symbol **A** stands for something like a bud here, and the first parameter, *t*, is a time delay: *t* is counted down, and a certain number of steps (here 6) must pass before a lateral branch starts growing. The second parameter, *k*, has only the values +1 or –1 and controls the orientation of the branch, similar to the tree example above, but not changing at random: *k* is systematically alternating between –1 and +1.

```
module A(int t, int k);
Axiom ==> L(100) A(0, 1);
A(t, k) ==>
         if (t > 0) ( A(t-1, k) )
         else
         ( F(1) [ RU(k*45) A(6, k) ] F(1) RU(3) A(0, -k) );
F(x) ==> F(1.15*x)
```

L-systems like this one naturally challenge the plant designer to explore their potential by playing around with parameters: E.g., if one reduces the initial delay in the branches from 6 to 2, branches will emerge earlier and a more compact form of the structure will result (Fig. 8b).

4.3 Interpretive rules

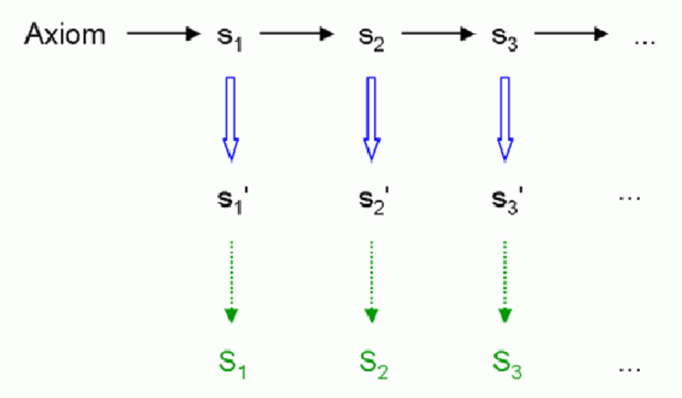
A very useful extension of the L-system formalism is an extra type of rules which are applied in a different manner: Whereas the “normal” L-system rules (also called *generative rules*) are iteratively applied to a string in order to obtain descriptions of new developmental stages, the so-called *interpretive rules* are applied only as a pre-processing for geomet-



a b

Figure 8: (a) Fern leaf produced by a parametric L-system (see text); (b) variant with reduced delay parameter for branch emergence

rical interpretation, and their application has no influence on the formation of the next developmental step:



In this diagram, the blue hollow vertical arrows represent the application of the interpretive rules, and dotted vertical arrows stand for the subsequent interpretation by the turtle. Particularly, the specification of graphical details of certain objects or organs, which are represented in the strings s_1, s_2, ... as a single symbol, can be given by an interpretive rule with this symbol as its left-hand side.[3] For example, in the following L-system the symbol **`A`** is copied 8-fold and shifted in the plane by a generative rule which is iteratively applied, whereas the interpretive rule transforms this **`A`** into a quadratic box. Both types of rules have to be separated in different “blocks” named **`run`** and **`interpret`**, and a command

---

[3] In the literature, interpretive rules were sometimes also called “homomorphisms”, but this is a misleading naming, because the usual, generative rules can mathematically also be seen as homomorphisms of a so-called free monoid; see, e.g., [Vitányi 1976].

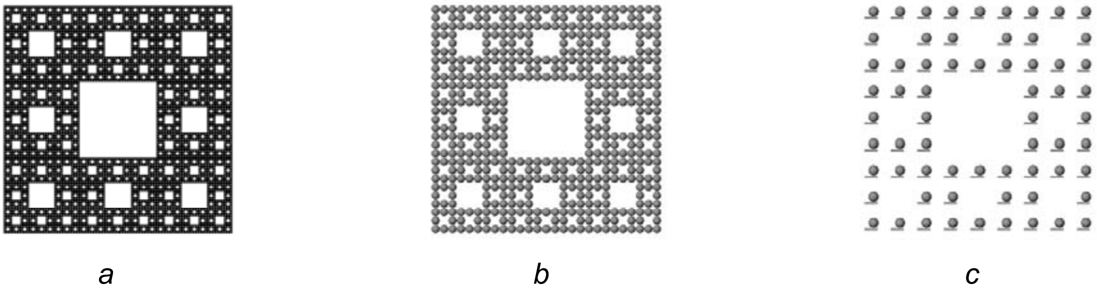


Figure 9: Different approximations of the Menger sponge fractal, obtained with different interpretive rules for the symbol **`A`** (see text)

"**`applyInterpretation`**" has to be given in order to apply the interpretive rules in the right moment:

```
public void run()
{
   [
   Axiom ==> A;
   A ==> Scale(0.3333) for (i:(-1:1))
                         for (j:(-1:1))
                           if ((i+1)*(j+1) != 1)
                             ( [ Translate(i, j, 0) A ] );
   ]
   applyInterpretation();
}
public void interpret()
   [
   A ==> Box;
   ]
```

The resulting pattern after 5 steps, approximating a so-called Menger sponge fractal, is shown in Figure 9a. The "Scale" command enforces a shrinking in every developmental step, to compensate for the 3-fold length of the result of copying.

If we now replace the interpretive rule by

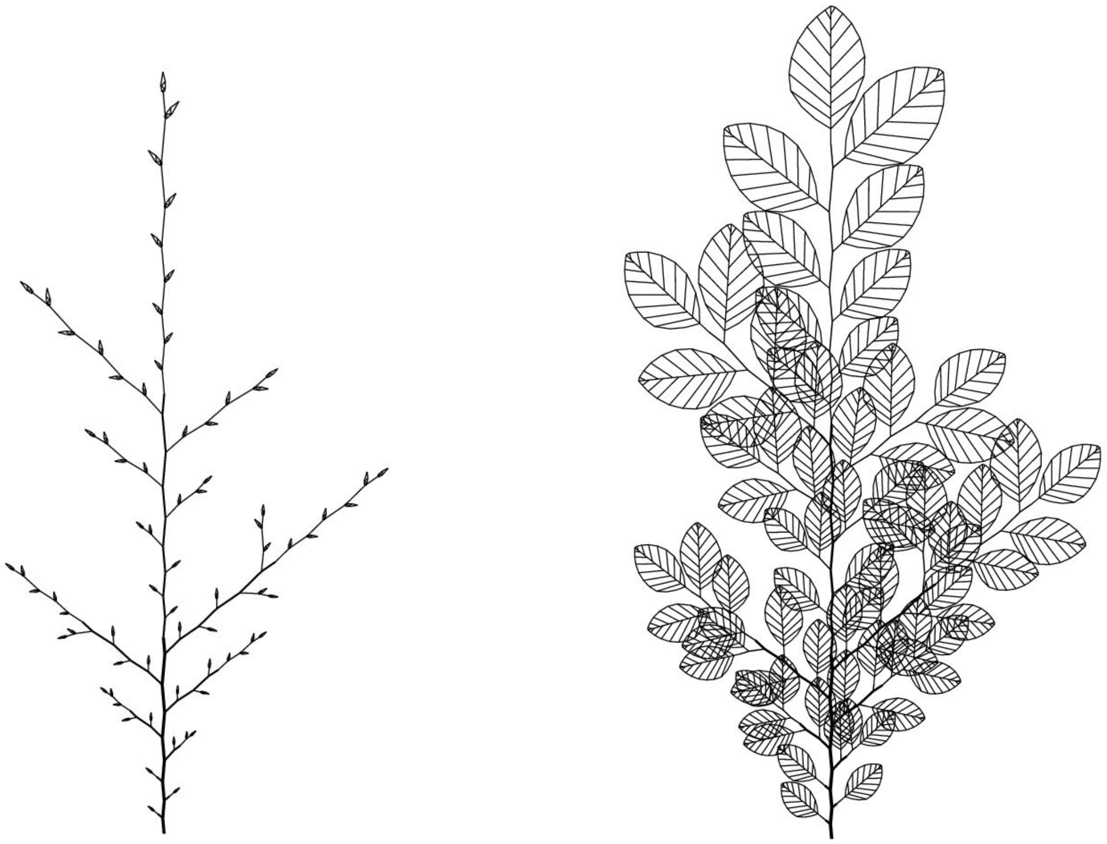
```
A ==> Sphere(0.5);
```

we get after 4 steps the result depicted in Fig. 9b. With the number of steps approaching infinity, the limit set will be the same fractal as in the first version. The same holds for the variant with

```
A ==> Box(0.1, 0.5, 0.1) Translate(0.1, 0.25, 0) Sphere(0.2);
```

which defines an arrangement of a flat box and a smaller sphere as initial configuration; the result after 3 steps is shown in Fig. 9c.

The right-hand side of an interpretive rule must not necessarily contain a command generating a geometrical body, like **`Box`**, **`Sphere`** or **`F`** (the latter making a cylinder). The following



a b

Figure 10: Model of a beech twig (left: in winter, right: in summer with the buds grown to leaves) based on an L-system; from [Kurth 1999]

example system, with an **RU** command on the r.h.s. of an interpretive rule, simulates a clock, with the correct ratio of revolvements of little and big hand (the hands modelled by **F** commands):

```
public void run()
{
   [
   Axiom ==> [ A(0, 0.5) D(0.7) F(60) ] A(0, 6) F(100);
   A(t, speed) ==> A(t+1, speed);
   ]
   applyInterpretation();
}
public void interpret()
   [
   A(t, speed) ==> RU(speed*t);
   ]
```

Interpretive rules considerably enhance the expressive possibilities of graphically-interpreted L-systems.

Using L-systems with the extensions introduced so far, it is already possible to create quite realistic-looking pictures of plants or twigs (Figs. 10, 11). Both models shown here are based on botanical observations and measurements and use only **F** commands for their

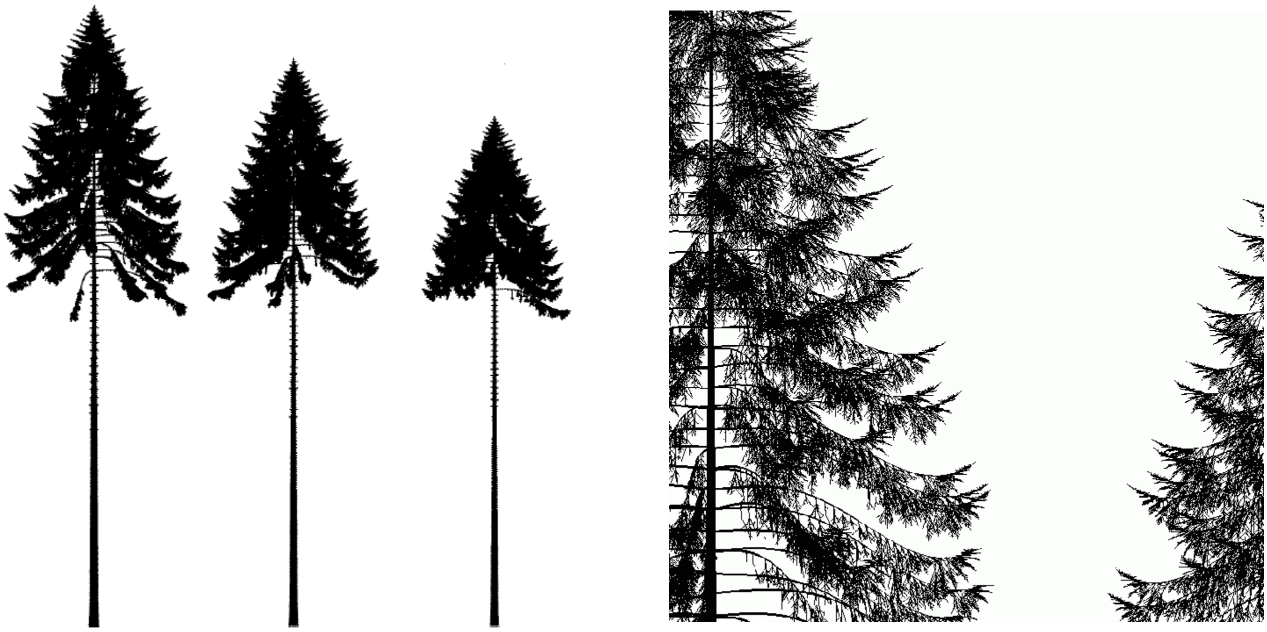


Figure 11: L-system-based model of spruce (Picea abies) trees from the Solling mountains Left picture: 3 representatives of tree classes (dominant, median, suppressed), right: zoom into two of the trees; from [Kurth 1999]

geometrical elements, which are in fact arranged in a virtual 3-D space (shown is only a parallel projection to a plane).

Although the trees from Figure 11 lack any surface details, colours or lighting and consist only of cylindrical elements, their patterns of branching are quite faithful to nature and allow their usage in simulation models of physical processes, e.g., water transport or distribution of sunlight in the canopy. Exemplarily, Figure 12 shows the resulting water potential profiles along selected branches in the crown of the virtual spruce tree shown on the left,

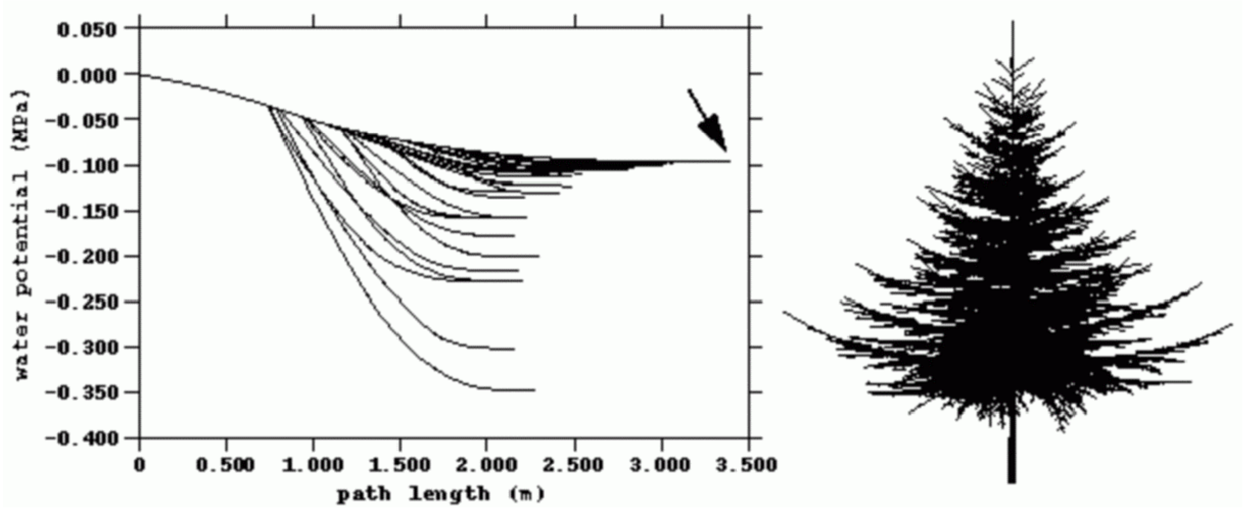


Figure 12: Virtual water potential profiles (left side) along selected branches of the virtual spruce tree (right side), obtained with the software tools Grogra and Hydra (from [Früh & Kurth 1999]).
Each line in the diagramme corresponds to a path from the tree base to a selected branch tip

Figure 13: Virtual Solling landscape, using rendered trees from L-systems and terrain data from a Geographical Information System. From [Knauft 2000]

when a flow simulation model based on differential equations is applied on the tree axes with their capacities and resistances [Früh & Kurth 1999].

On the other hand, when the geometrical elements of the virtual plants are rendered using standard computer-graphics techniques, the trees can be copied and arranged in visualized virtual landscapes like in Figure 13. Here, an interface programme taking terrain data from a GIS (Geographical Information System) and an additional algorithm for the creation of realistic planting patterns of trees were used; see [Knauft 2000].

4.4 Context-sensitivity

All the L-systems shown above allow only a flow of information from the predecessor (in a rule) of a symbol to the symbol itself (“lineage control”). However, in nature we have often the situation that growth or development of an organ is influenced by some information (signals, energy flow, substances) coming from other parts of the existing structure. If we assume that this information comes from the neighbourhood (in a topological sense) of the organ under consideration, it is possible to model such influences by *context-sensitive L-systems*: Applicability of a rule is restricted to the cases when a certain predefined context surrounds the symbol given on the left-hand side of the rule. This context is again specified by symbols, which must be present to the left or to the right of the given symbol in the string representation of the generated structure.[4] Using this formalism, the transport of a signal or of a substance through a growing or static structure can be simulated. Let us consider the following L-system:

```
module A(int age);
module B(super.length, super.color) extends F(length, 3, color);
Axiom ==> A(0);
A(t), (t < 5) ==> B(10, 2) A(t+1);
A(t), (t == 5) ==> B(10, 4);
```

---

[4] To be precise, we allow several neighbours to the right in the case of branching: The basic element of each branch emerging in x is considered as a neighbour of x. Furthermore, we permit the skipping of pairs of brackets [...] during checking the context conditions.

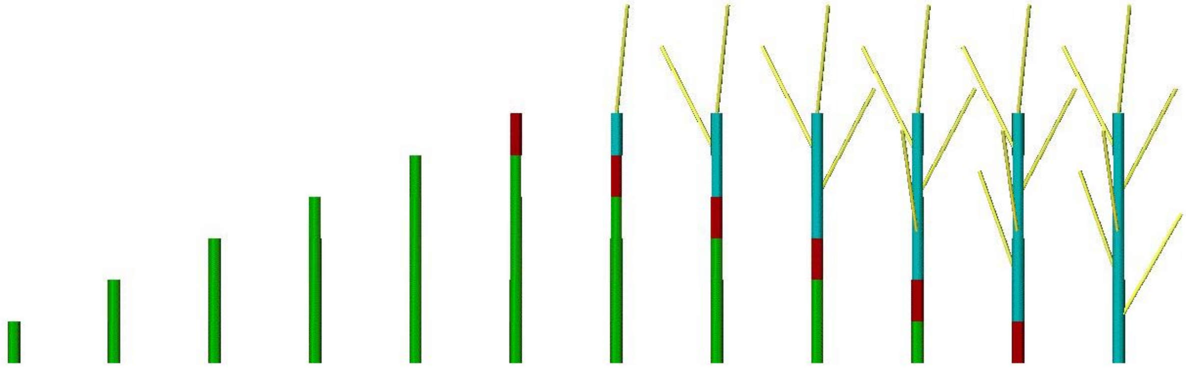


Figure 14: Signal propagation modelled by a context-sensitive L-system (see text)

```
6   B(s, 2) (* B(r, 4) *) ==> B(s, 4);
7   B(s, 4) ==> B(s, 3) [ RH(random(0, 360)) RU(30) F(30, 1, 14) ]
```

In line 2, `B` is defined to symbolise a cylinder of diameter 3 and of arbitrary length and colour. Symbol `A` has the meaning of a bud, which produces cylindric stem segments `B(10, 2)` of length 10 and colour 2 (green) while ageing (`A(t)` becomes `A(t+1)`) in line 4. When it reaches age 5, it is transformed in a red segment (`B(10, 4)`) and stops growing (there is no `A` on the right-hand side of the rule in line 5). The rule in line 6 is the context-sensitive one: It waits for a red segment (context `B(r, 4)`, enclosed in `(* ... *)`) to occur to the right (geometrically: above) a green segment. If this happens, the green segment is itself

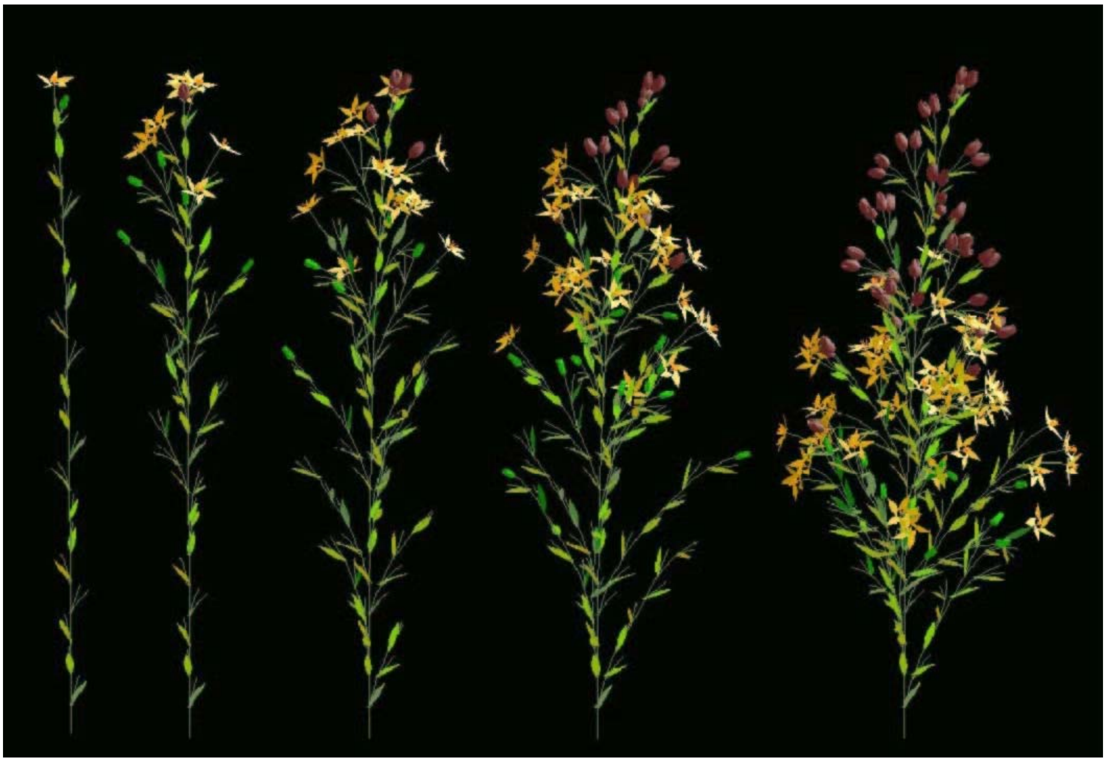


Figure 15: Simulation of flower development of the plant Mycelis muralis, obtained from a context-sensitive L-system. From [Prusinkiewicz & Lindenmayer 1990, p. 91]

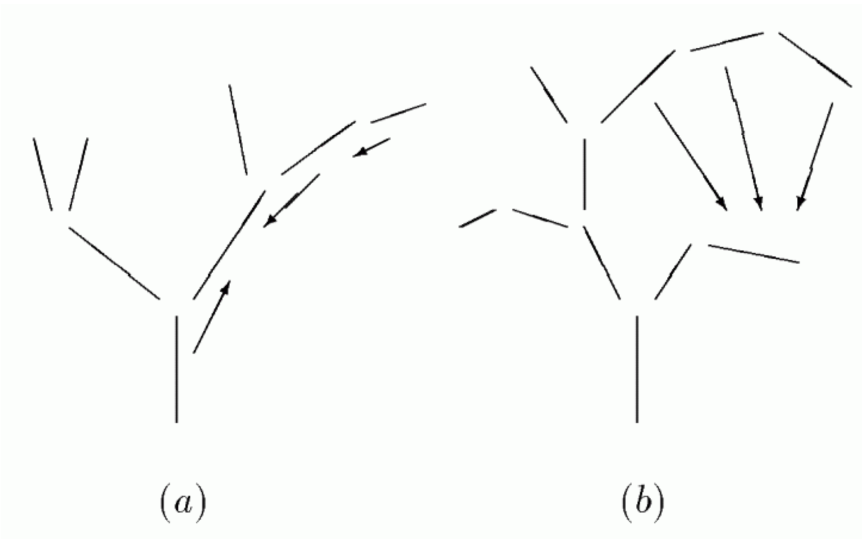


Figure 16: Local (a) and global (b) interactions in a geometrical structure representing some organism. Far-reaching effects (b), like shadowing, cannot be modelled by context-sensitive L-systems

replaced by a red one (`B(s, 4)` on the right-hand side). The last rule tells us that a red segment is in the next step always transformed into a blue segment (`B(s, 3)`) with a long, thin yellow branch (`F(30, 1, 14)`) in random direction. The development of this simple structure in 12 steps, with the red cylinders indicating the downward movement of the branch-inducing signal within the virtual plant, is traced in Figure 14.

Our example was very simplistic, but the same formalism can be used to simulate realistic hormonal signals and induction of flowering in rendered virtual plants (Fig. 15).

4.5 Global sensitivity

Interaction in the real world does not only take place between objects that are immediate neighbours. E.g., in a tree, information can pass from a stem segment to a neighbouring segment in the form of hormones or other substances (Fig. 16a), but also from segments that are far away, by shadowing (Fig. 16b).

Context-sensitive L-systems consider only a context in the sense of the string representation of the generated geometrical structure. This is not enough for modelling the behaviour of “globally sensitive” organs, which, e.g., react to shadow and can be influenced by parts of the structure that are in a far distance. For this reason, Prusinkiewicz et al. [1994] introduced “environmentally-sensitive L-systems”, which were later generalised by Měch & Prusinkiewicz [1996] under the name “open L-systems”. Independently, Kurth [1994] introduced “sensitive growth grammars”.[5] Common to all these approaches is the possibility of communication between distant entities or “modules” by the use of special “communication

---

[5] – which are not identical with the “relational growth grammars” described below in this paper.

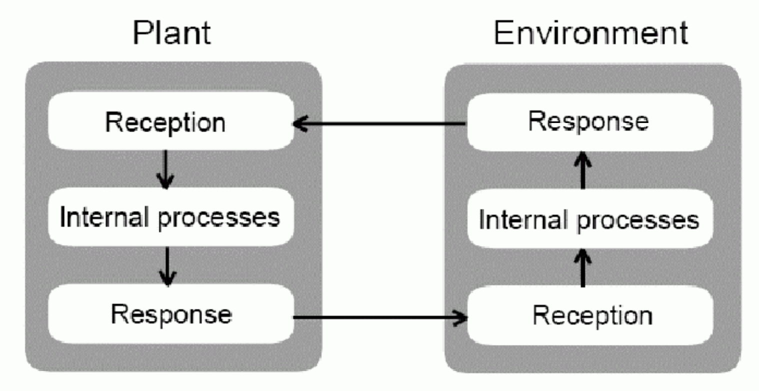


Figure 17: Division between models of an organism and of its environment according to Měch & Prusinkiewicz [1996]

modules" or "sensitive functions". Specific for the approach followed by Prusinkiewicz et al. is a strict conceptual separation maintained between the simulated part (represented by strings) and its "environment" (with the created geometrical structure as a part thereof). Both parts are differently modelled, and information exchange between the two simultaneously running simulations is mediated by special interfaces, the above-mentioned communication modules (Fig. 17).

In contrast, we try in our approach to simulate organisms and their environment in a uniform manner and using the same language XL. We feel that the border between organism and environment is in many cases somehow artificial. E.g., the shadowing parts in Figure 16 are at the same time parts of the virtual plant and of its virtual environment.

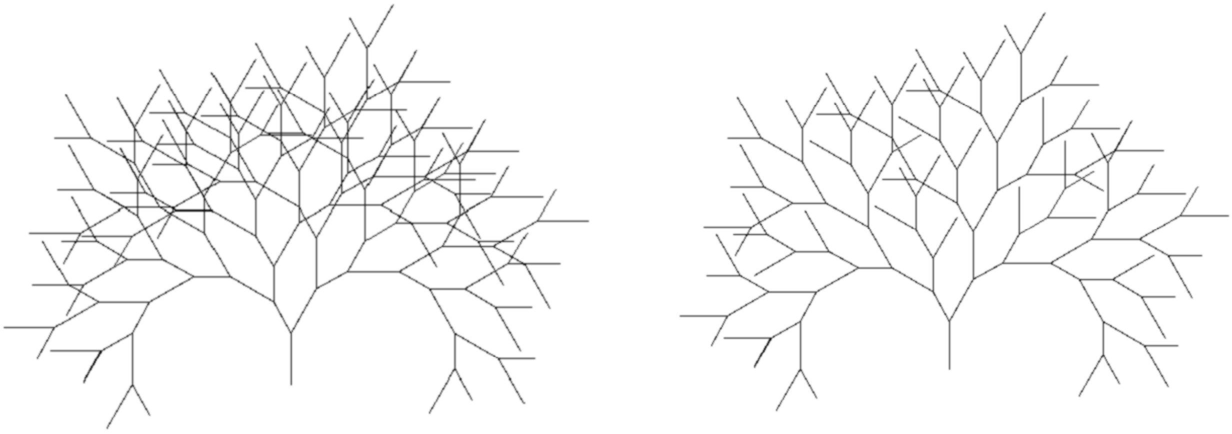
An example of a globally-sensitive L-system realized in our language XL is given below. It simulates "density-sensitive" buds that produce new shoots only if there is no other object closer than 60 length units.[6] To make the structure not too symmetrical, two different shoot types `F(100)` and `F(70)`, the latter being shorter, are used. The bud is named `A` and carries the information about the length of the shoot which it will produce in the next step as its parameter:

```
module A(int s);
Axiom ==> F(100) [ RU(-30) A(70) ] RU(30) A(100);
a:A(s) ==> if ( forall(distance(a, (* F *)) > 60) )
           ( RH(180) F(s) [ RU(-30) A(70) ] RU(30) A(100) )
```

The first rule creates initially a long shoot with two buds, A(70) and A(100), at its tip. In the second rule, the bud A(s) on the left-hand side is *labelled* by a name, a, to enable referencing on the right-hand side to this particular bud. In the "if"-condition on the right-hand side, we find a *query function*, "forall", which looks for all objects of type "F" (specified by "(* F *)") and checks their Euclidean distance to bud a. Only if all these distances exceed 60 length units, the rule is applied and the bud is replaced by a new shoot (F(s)) with

---

[6] Notice that the "context condition" is purely geometrically defined and does not require that the potential obstacles are topological neighbours of the bud, i.e. that they are directly connected with it.



a

b

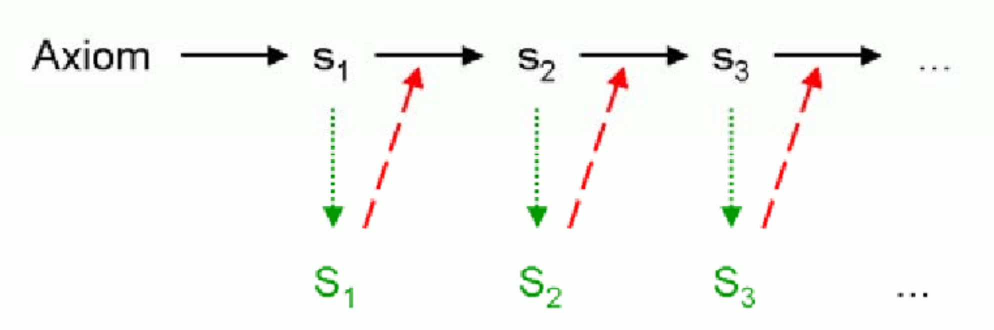
Figure 18: Simple tree with dichotomous branching after 7 developmental steps, generated by a grammar (a) without and (b) with a condition which incorporates global sensitivity (here: suppression of growth by close other objects); see text (adapted from [Kurth 1994].)

60 length units, the rule is applied and the bud is replaced by a new shoot (F(s)) with two buds at its end (last line). The search is done exhaustively in the whole created structure here.

If we omit the "if"-condition, the result of this L-system is just a binary tree with exponential growth, as shown in Figure 18a. With sensitivity in action, not all buds continue growing, and the resulting structure contains fewer branches and fewer crossings between them (Fig. 18b). Notice that not all crossings of branches are eliminated: The reason is that the emptiness of the geometrical neighbourhood of a bud is checked *before* all the new branches have grown. It can happen that closeness or even crossing occurs through simultaneous growth of two shoots whose buds were not close enough before, with the consequence that they did not stop to grow.

With similarly simple grammars, competition between several virtual plants for space and light can be simulated (Fig. 19; code not shown).

To condense the effect of global sensitivity again in a diagram, we find that the currently produced structure S_i can exert influence on the application of generative rules that rewrite the string s_i to s_{i+1} (red, broken arrows):



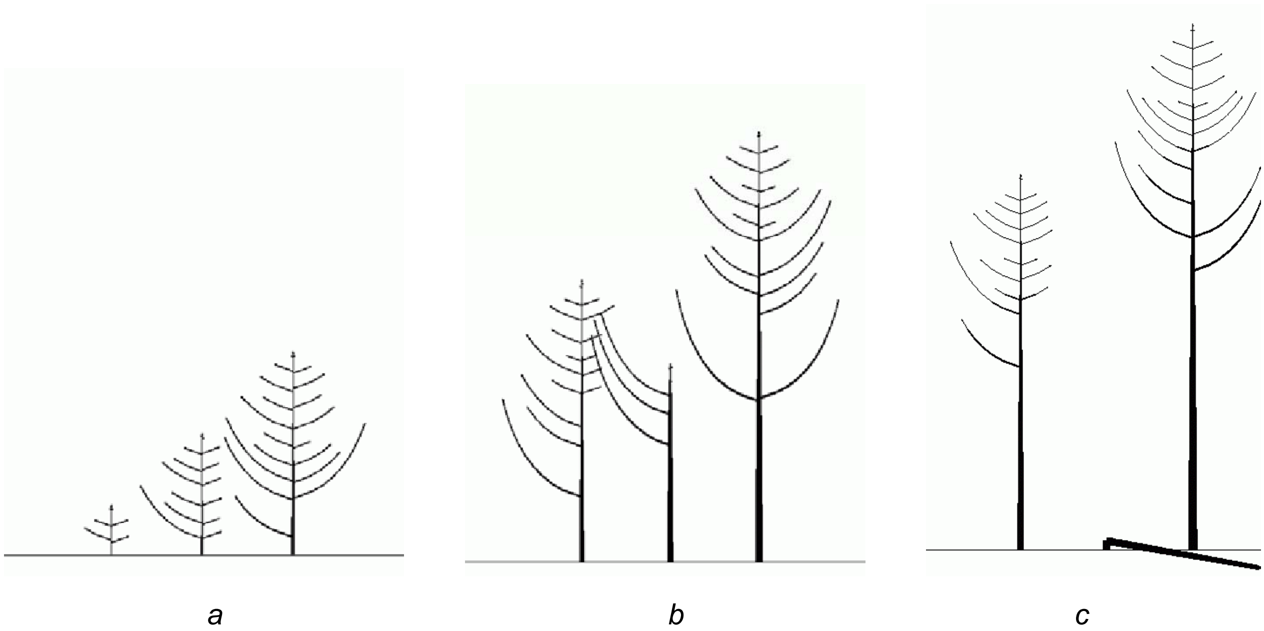


Figure 19: Growth of three virtual trees competing for light, modelled using a globally-sensitive grammar; from [Kurth 1999]

Of course, it is possible to combine this information flow with the use of interpretive rules (see above).

5 Relational growth grammars

L-systems have been very fruitful for theoretical investigations in formal language theory and for creating realistic-looking models of plants. However, even if all the above-presented extensions are included, they have some limitations:

- In interpreted L-systems (with turtle geometry and with brackets for branching), only two possible relations can be created between the simulated objects: **A** can be a *direct successor* of **B** or can be supported by **B** as a *branch*. In reality, much more sorts of relations between objects are possible and can be worth modelling.
- L-systems are not really an appropriate tool for the creation of truly 2-dimensional or even 3-dimensional arrangements, like tessellations in the plane or cellwork systems (e.g., in tissues). In fact, there exist formalisms like “map L-systems” and “cellwork L-systems” (see [Prusinkiewicz & Lindenmayer 1990]), but their definitions and usage are rather complicated. The reason is that the classical interpretation of bracketed strings by the turtle can only yield locally one-dimensional topologies that are homeomorphic to trees. Particularly, *cycles* and *networks* can be created only if additional tools or tricks are allowed.
- *Multiscaled modelling*, i.e., the simultaneous specification of some structure at several different levels of resolution, is not supported.

- For the biologists, it is a drawback that *genotype* and *phenotype* of an organism cannot be modelled in the same formal framework (although the DNA molecule has basically string structure).
- From the perspective of software development, L-systems as a programming language are a poor language; particularly, the *object-oriented* programming (OOP) style, which is today very commonly used by programmers, is not supported: The fundamental units of the formalism are only symbols (perhaps with some added numerical parameters), no objects in the sense of OOP. Particularly, no hierarchy of object classes, where specialised classes inherit properties from more general classes, can be defined in the classical L-system formalism.

These were reasons enough to design a new formalism, “relational growth grammars” (RGG), and a corresponding programming language, XL (eXtended L-systems language). An RGG is a rewriting system operating on *graphs* instead of strings – here, a graph is a structure consisting of nodes and arcs (also called “edges”) connecting some of these nodes, and it can have cyclic substructures. We speak of “relational” grammars because we permit several types of edges (relations). This extension of the L-system concept addresses the first 4 points above [Kniemeyer et al. 2004]. The fifth point is addressed by permitting RGG rules as constructions in a programming language (XL), which is at the same time a true extension of the object-oriented language Java, and by permitting Java objects as nodes of the graphs that are rewritten.[7] An exact mathematical definition of RGG and a precise language specification for XL will be given by Kniemeyer [2007].

The graphs which are rewritten by an XL programme can also be seen as generalisations of scene graphs, as they are known from 3-D modelling languages and tools like VRML, Java 3D or Maya. Particularly, their nodes can stand for geometrical objects and also for transformations of objects (like translation, rotation, scaling...). Indeed, we have already used this feature in our Menger sponge example above (see Fig. 9).

The general structure of an RGG rule is shown in Figure 20. An RGG is composed of such rules, which are usually applied to a given graph in parallel, like L-system rules.

The application of a simple RGG rule to a given graph is demonstrated in Figure 21. Here, the upper part of the Figure describes the rule. There is no context *C*, no condition *E* and no procedural code *P* in this case. So, the left-hand side, two nodes of classes `A` and `B` that are connected by a directed edge from `A` to `B`, has to be replaced by the right-hand side wherever it occurs. There are two sorts of edges (relations) in this example, which are visualised as solid and dotted arrows, respectively. The lower part of the Figure shows exemplarily an application of this rule: the red part on the left, encircled by a solid blue line, is

---

[7] A similar approach led to the language “L+C” [Karwowski & Prusinkiewicz 2003], which is an extension of C++ by L-system rules, but this language does not include graph transformations.

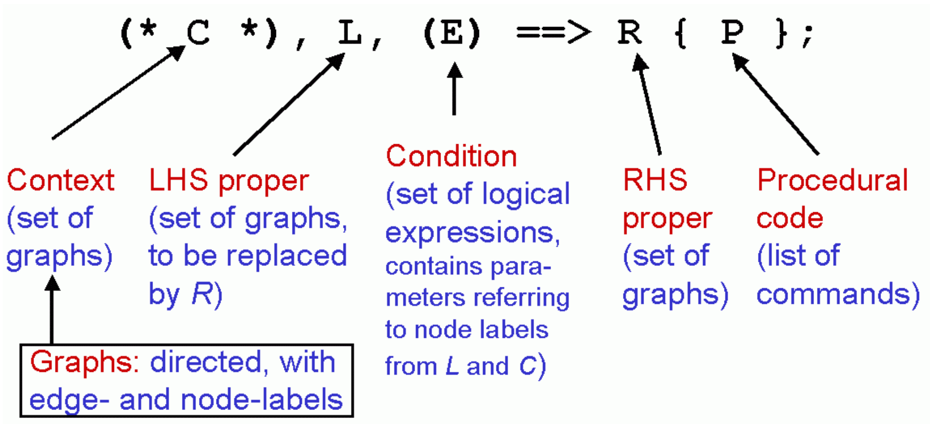


Figure 20: Syntactic structure of an RGG rule
The essential effect of this rule is to replace L by R (and to execute P)

identical with the left-hand side of the rule and is thus replaced by the corresponding right-hand side (result: lower right part of the Figure). Notice that the left-hand side of the rule does not match the part of the graph that is encircled by the broken blue line, because the edge connecting **A** with **B** is of the wrong sort there.

Relational growth grammars are a special form of *graph grammars*. As for L-systems, there exists a well-developed theory about graph grammars [Rozenberg 1997]. L-systems can be subsumed as a special case, because strings can be represented as special graphs with a linear structure, with edges of a certain, fixed type "successor" between consecutive symbols. In XL, edges are generally written down in the form "—*edgelabel*—>", where "*edgelabel*" specifies the type of the edge – but because the edge type "successor" is so often used, a simple blank symbol is allowed instead of "—**successor**—>". This

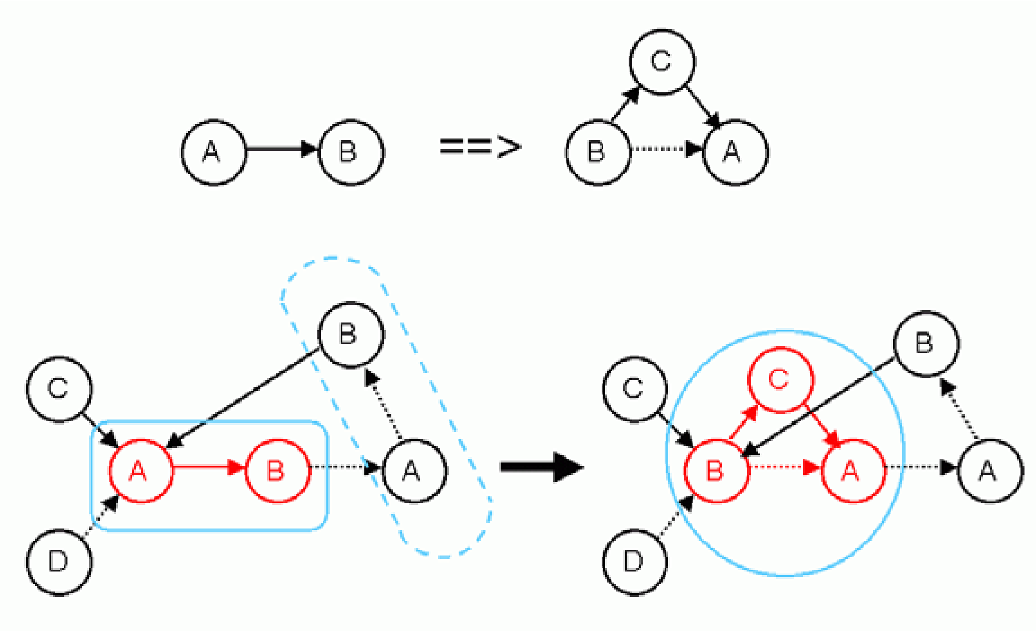


Figure 21: Application of a relational growth grammar rule (upper part) to a graph (lower part)

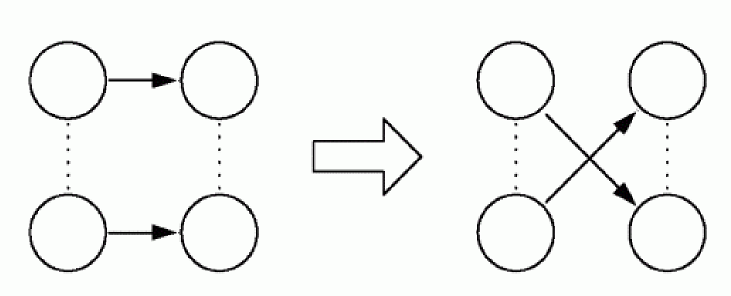


Figure 22: Graphical representation of an RGG-rule for genetic crossing-over Unbroken arrows stand for the successor relation in base sequences of DNA, dotted lines for alignment between two homologous DNA strings

convention allows us to write down L-system rules in XL in a quite familiar manner – and in fact, all L-system examples shown above were directly taken from XL programmes. In order to make them readable by an XL compiler (like that in the software GroIMP, see below), one has only to enclose the rules (not the "module" declarations) in a surrounding construction of the form

```
public void run()
   [
   ...
   ]
```

(with the exception of the examples using interpretive rules, where a similar construction was already explicitly given). The reason is that RGG rules in XL can be organised in several blocks, in order to enable a better control of the order of rule application – thus making accessible the possibilities of so-called table L-systems [Rozenberg 1973].

However, the capacity of RGGs goes far beyond L-systems. A simple example for a graph transformation which cannot be expressed as an L-system rule occurs in genetics: In the context of sexual reproduction, there is the process of recombination of genetic information, which takes place by so-called "crossing over" of two aligned DNA strings. The basic transformation, which exchanges the bindings between the two DNA strings, is shown in Figure 22.

An XL representation of this rule is

```
a b, c d, (* a -align- c *) ==>> a d, c b;
```

and in fact we have used this rule together with one for mutation to simulate the evolution of artificial "biomorphs" [Kniemeyer *et al.* 2004].

In addition to the genetic level, it is also possible to represent biochemical reactions and metabolic reaction networks in the form of RGG rules. We will not go into details here (see [Buck-Sorlin *et al.* 2005]), but we show some of the visual results of models which have as a non-visible part also some metabolic and, in some cases, also genetic components: Fig-

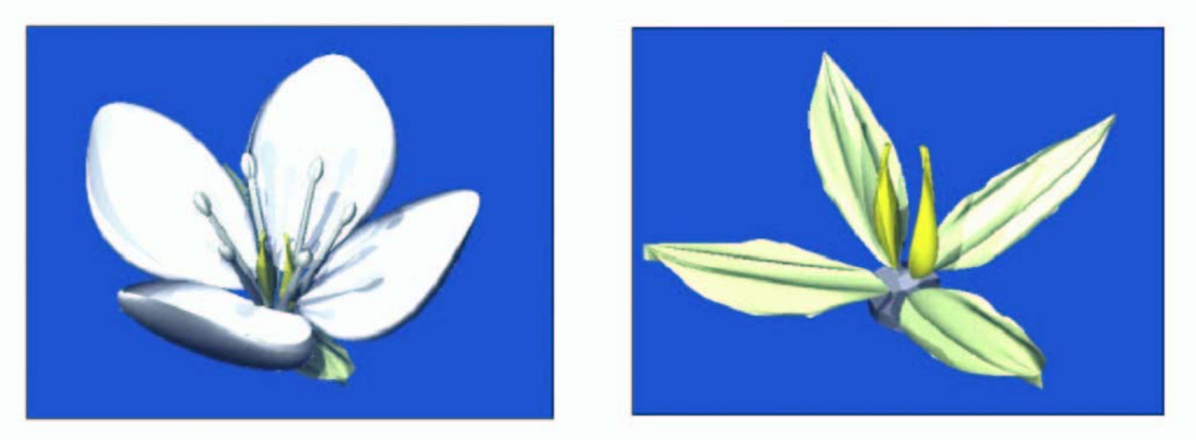


Figure 23: Virtual "wild type" (left) and "mutant" (right) of a flowering plant, generated by an RGG which encodes also the causal genes and (a part of) the mediating transcription-factor reaction network (from [Kniemeyer et al*. 2004], based on earlier work by [Kim 2001])*

ure 23 shows a flower and a mutant thereof, where some gene is deactivated – the effect of the "silent gene", namely, the lack of petals, is mediated by a reaction network (see [Kniemeyer *et al.* 2004]).

Figure 24 shows two developmental steps of a virtual rapeseed plant, which is assimilating virtual carbon depending on virtual light interception and nitrogen availability, and which allocates this carbon to its growing organs according to the (time-dependent) relations between source and sink strengths; see [Groer 2006]. Figure 25 shows a virtual barley plant (including the root system, which was not modelled in the other examples shown above), which depends in its growth not only on sunlight, but also on a reaction network producing a plant hormone (Gibberellic acid), like in real plants, and which can reproduce and mutate (Buck-Sorlin *et al.,* partially unpublished work, see also [Buck-Sorlin *et al.* 2005]).

In the field of computer-graphical modelling of plants, the traditional L-system approach has recently been challenged by the Xfrog software, developed by Deussen and Lintermann, see [Deussen 2003]: Here, graphs are interactively edited which define implicitly rules for the multiplication and arrangement of geometrical objects. Although this graph-controlled approach is not based on biological laws, it allows a quick interactive specification of complex vegetation models. However, the graphs used in Xfrog and the creation of geometrical structures based on them can exactly be reproduced in the language XL (if RGG rules are complemented by a further type of rules, so-called instantiation rules) – see [Henke 2007]. The relations between Xfrog and our rule-based approach will be subject of a forthcoming article [Henke *et al*. 2007]. Figure 26 shows results of the simulation of Xfrog-defined structures in XL.

Figure 24: Virtual rapeseed, generated by an RGG taking photosynthesis, nitrogen uptake and carbon allocation into account, all programmed in XL. From [Groer 2006]

But the use of RGGs is not restricted to plants. Figure 27 indicates other fields of application, which are not yet completely explored.

Relational growth grammars, embedded in XL programmes, can be read and executed by a software named *GroIMP* (Growth-grammar related Interactive Modelling Platform). This platform-independent software contains an XL compiler, a development tool (extended editor) for XL, a 3-D modeller and renderer (including a raytracer), a 2-D visualiser for the transformed graphs, windows for plotting functions, editing facilities for 3-D objects and for their attributes, tools for generating textures, networking facilities, a collection of RGG examples and a tutorial for the language XL. XL and GroIMP will be thoroughly documented in [Kniemeyer 2007].[8] A screenshot of the current GroIMP version is shown in Figure 28. All images of virtual structures in this paper were generated with GroIMP, with the following exceptions: Figures 10 to 12 and 19 were created with the GroIMP-forerunner *Grogra* [Kurth 1994], 13 is from [Knauft 2000], 15 is from [Prusinkiewicz & Lindenmayer 1990].

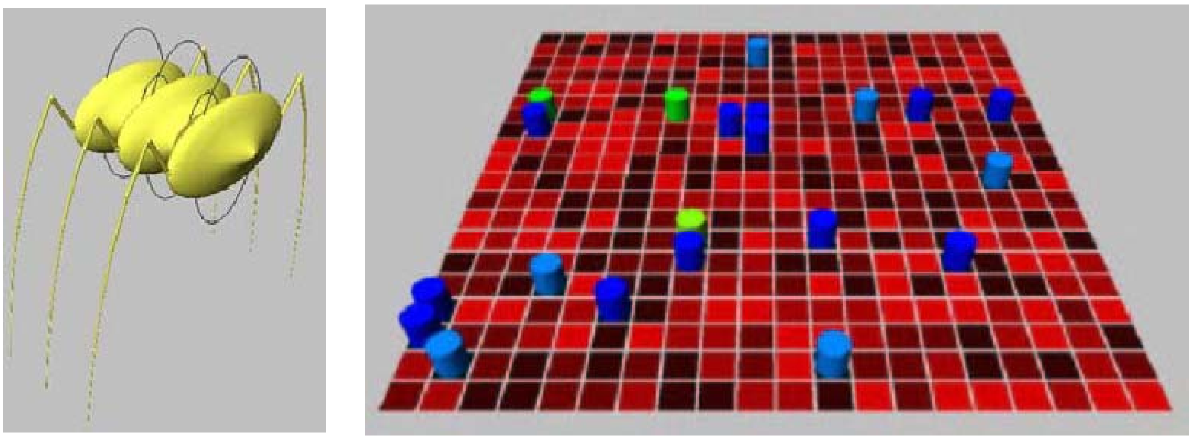
---

[8] The software is available by download under the GNU public licence (GPL), i.e., as an open-source tool; see http://www.grogra.de .

Figure 25: Virtual barley plant with hormonal metabolism and genetic features, see text. From [Buck-Sorlin et al. *2006]*

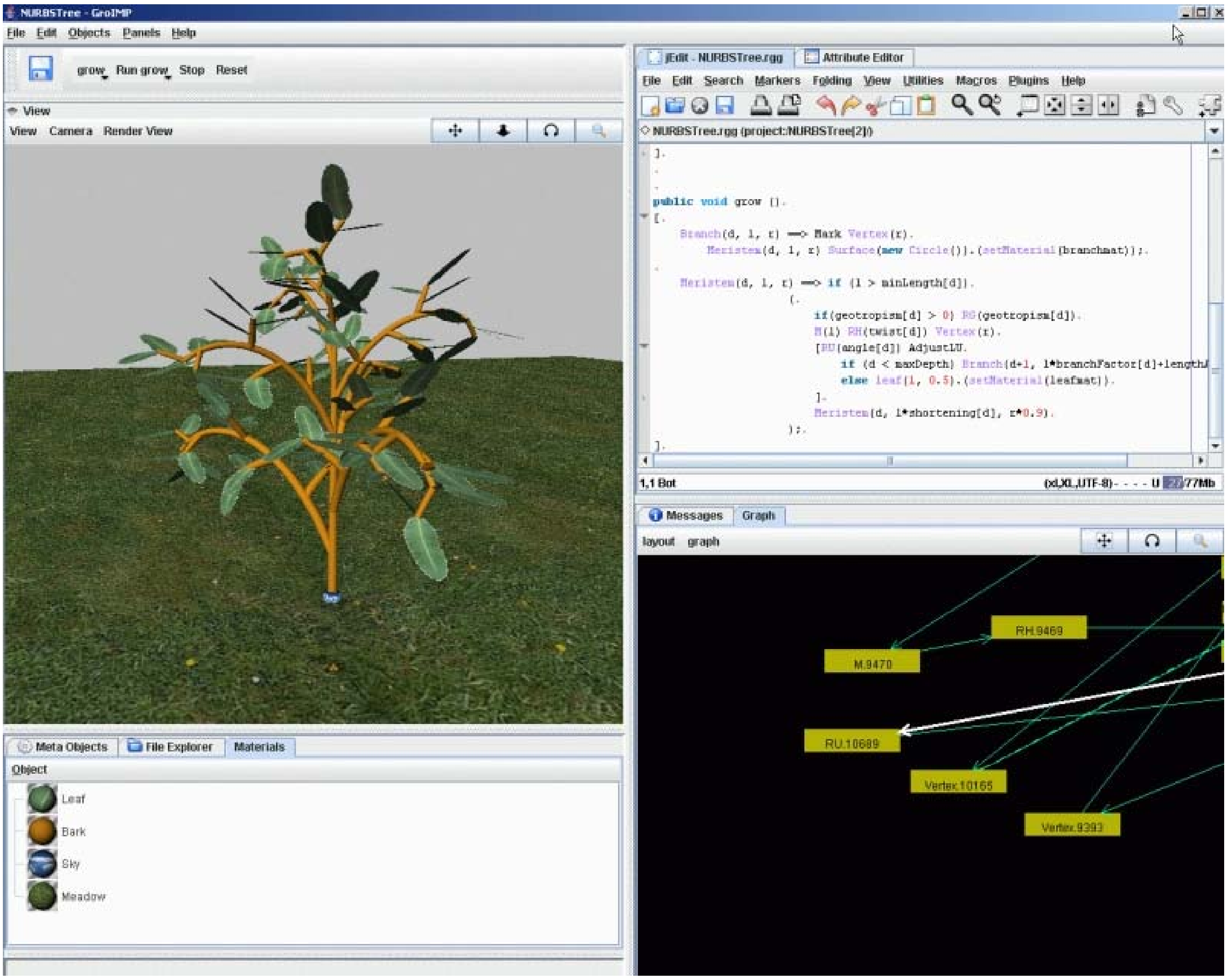
a *b*

Figure 26: Virtual plants generated using instantiation rules in XL that simulate the way in which the Xfrog software [Deussen 2003] specifies virtual plants; from [Henke 2007]



a b

Figure 27: RGG-based modelling beyond plants. (a) Insect-like animal (Bischof, unpublished). (b) Simulation of the agent-based “Sugarscape” model of an artificial society on a rectangular grid; from [Graeber 2006]



```
].


public void grow ()
[
    Branch(d, l, r) ==> Mark Vertex(r)
        Meristem(d, l, r) Surface(new Circle()).(setMaterial(branchmat));

    Meristem(d, l, r) ==> if (l > minLength[d])
                    (
                        if(geotropism[d] > 0) RG(geotropism[d])
                        M(l) RH(twist[d]) Vertex(r)
                        [RU(angle[d]) AdjustLU
                            if (d < maxDepth) Branch(d+1, l*branchFactor[d]+length
                            else leaf(l, 0.5).(setMaterial(leafmat))
                        ]
                        Meristem(d, l*shortening[d], r*0.9)
                    );
].
```

Figure 28: Screenshot of the GroIMP software (download possible from www.grogra.de)

6 Discussion

The visual models of plants obtained with L-systems and relational growth grammars corroborate the assumption that these mathematical formalisms are appropriate tools to capture essential aspects of *morphological* structures in the world of *plants* – maybe analogously to the appropriateness of differential equations for modelling phenomena in physics. In fact, it is straightforward to formalise such morphological phenomena which are known by botanists under the notions of “acrotony”, “neoformation”, “sleeping buds”, “reiteration”, “apical control” etc. in the language of L-systems (see, e.g., [Kurth 1996]). If L-systems are extended to more flexible formalisms like relational growth grammars, this appropriateness can also be stated for plant models that connect several levels of spatial resolution, and for *functional-structural models* where the purely morphological layer is complemented by processes taking place “behind” the visual world or at smaller scales. It can be conjectured that models for the evolution of *network structures* could also profit from a formalisation in the frame of a rule-based language like RGGs. Early studies did already explore some non-biological applications of L-systems: Specification of planar tilings, music [Prusinkiewicz & Hanan 1989], ornaments, weave patterns, architecture, evaluation of mathematical expressions, robotics [Goel & Rozehnal 1991]. In a present students’ course at the University of Technology at Cottbus, we just explore the usage of RGGs in architectural design.

Another possible field of applications for rule-based formalisms is chemistry. Chemical reactions have some similarity with grammar rules, but they usually take place in an unstructured “soup” consisting of a very large number of molecules – hence the linear ordering which we had in the L-system strings does not apply here, and it is doubtful if one can speak of morphological structures in this case, except when we restrict our focus to single, but complex molecules. Another feature of L-systems and RGGs that make them seem inappropriate for applications in chemistry and physics is their discretisation of time. Mathematical descriptions of classical dynamical systems make use of the concept of continuous time. This concept does also make sense when smooth animations of growth processes, of animal movement etc. are wanted. However, it has already been shown that “timed L-systems” can be defined, which abandon the concept of fixed-length developmental steps in favour of continuous growth and event-driven application of rules [Prusinkiewicz & Lindenmayer 1990]. The incorporation of these modifications into more advanced formalisms like RGGs is still to be done, but will probably pose no great difficulties.

A probably even more urgent need for theoretical and practical research is revealed by the question how truly 2-D and 3-D structures like planar maps or spatial cellworks – in contrast to essentially 1-dimensional tree-like structures – and their growth and dynamics can be elegantly modelled using an appropriate grammar formalism. Until this challenge is not resolved by a really intuitive and compact calculus, we cannot say that true *picture mor-*

phology can be satisfactorily modelled by known rule-based formalisms like L-systems. However, our experience from the creation of virtual plants and of some other interesting virtual patterns suggests that there are some features inherent to the rule-based programming paradigm which make it a promising candidate for playing a prominent role in a future theory of picture morphology.

Acknowledgements

Ole Kniemeyer designed and implemented the programming language XL and the GroIMP software; without these tools and without his help this study would not have been possible. Gerhard Buck-Sorlin, Reinhard Hemmerling, Michael Henke, Christian Groer, Bernd Graeber, Michael Tauer, Branislav Sloboda, Thomas Früh, Helge Dzierzon, Dirk Lanwert and Falk-Juri Knauft provided examples for applications, stimulated discussions and helped to improve certain aspects of the above-mentioned tools. Research was partially funded by the Deutsche Forschungsgemeinschaft and by the German Ministry of Research and Technology. All support is gratefully acknowledged.

References

Abelson, H. & diSessa, A. A. (1982): *Turtle Geometry*. Cambridge: MIT Press.

Ascher, Marcia (2003): Die kolam-Figuren Südindiens. *Spektrum der Wissenschaft*, 6/2003: 74–80.

Buck-Sorlin, Gerhard & Kniemeyer, Ole & Kurth, Winfried (2005): Barley morphology, genetics and hormonal regulation of internode elongation modelled by a Relational Growth Grammar. *New Phytologist*, 166 (3): 859–867.

Buck-Sorlin, Gerhard & Hemmerling, Reinhard & Burema, Benno & Kniemeyer, Ole & Kurth, Winfried (2006): Towards “Virtual Barley”: Relational Growth Grammars as a universal framework for rule-based hierarchical modelling of morphogenesis, plant function, regulatory networks, and genetic processes. Waterman-Seminar, IPK Gatersleben, 15. 12. 2006 (unpublished ppt-presentation).

Deussen, Oliver (2003): *Computergenerierte Pflanzen – Technik und Design digitaler Pflanzenwelten*. Berlin: Springer.

Früh, Thomas & Kurth, Winfried (1999): The hydraulic system of trees: Theoretical framework and numerical simulation. *Journal of Theoretical Biology*, 201 (4): 251–270.

Goel, Narendra S. & Rozehnal, Ivan (1991): Some non-biological applications of L-systems. *International Journal of General Systems*, 18 (4): 321–405+color plates.

Graeber, Bernd (2006): Vergleich von Java und XL anhand des Agentenmodells “Sugarscape”. Studienarbeit, BTU Cottbus, Lehrstuhl Praktische Informatik / Grafische Systeme.

Groer, Christian (2006): Dynamisches 3D-Modell der Rapspflanze (Brassica napus L.) zur Bestimmung optimaler Ertragskomponenten bei unterschiedlicher Stickstoffdüngung. Diploma thesis, BTU Cottbus (104 pp.).

Henke, Michael (2006): Entwurf und Implementation eines Baukastens zur 3D-Pflanzenmodellierung in GroIMP mittels Instanzierungsregeln. Diploma thesis, BTU Cottbus (156 pp.)

Henke, Michael & Kniemeyer, Ole & Kurth, Winfried (2007): Realisation and extension of the Xfrog approach for plant modelling in the graph-grammar based language XL. (in preparation)

Karwowski, R. & Prusinkiewicz, P. (2003): Design and implementation of the L+C modeling language. *Electronic Notes in Theoretical Computer Science*, 86 (2), 19 pp. (http://algorithmicbotany.org/papers/l+c.tcs2003.pdf).

Kim, Jan T. (2001): A generic formalism for modelling regulatory networks in morphogenesis. In: Kelemen, J. & Sosík, P. (eds.), *Advances in Artificial Life*. LNAI 2159. Berlin: Springer, 242–251.

Knauft, Falk-Juri (2000): Entwicklung von Methoden zur GIS-gestützten Visualisierung von Waldentwicklungsszenarien. Ph.D. thesis, Universität Göttingen, Fakultät für Forstwissenschaften und Waldökologie (126 pp.)

Kniemeyer, Ole & Buck-Sorlin, Gerhard & Kurth, Winfried (2004): A graph-grammar approach to Artificial Life. *Artificial Life*, 10 (4): 413–431.

Kniemeyer, Ole (2007): Design and implementation of a graph-grammar based language for functional-structural plant modelling. Ph.D. thesis, BTU Cottbus (forthcoming).

Kurth, Winfried (1994): Growth Grammar Interpreter GROGRA 2.4 – A software tool for the 3-dimensional interpretation of stochastic, sensitive growth grammars in the context of plant modelling. Introduction and Reference Manual. *Berichte des Forschungszentrums Waldökosysteme*, B, 38, Göttingen (192 pp.).

Kurth, Winfried (1996): Elemente einer Regelsprache zur dreidimensionalen Modellierung des Triebwachstums von Laubbäumen. In: Hempel, G. (ed.), Deutscher Verband Forstlicher Forschungsanstalten, Sektion Forstliche Biometrie und Informatik, 8. Tagung, Tharandt / Grillenburg, 25.-28. 9. 1995; Ljubljana: Biotechnische Fakultät, 174–187.

Kurth, Winfried (1999): *Die Simulation der Baumarchitektur mit Wachstumsgrammatiken*. Berlin: Wissenschaftlicher Verlag Berlin (327 pp.).

Lindenmayer, Aristid (1968): Mathematical models for cellular interactions in development. *Journal of Theoretical Biology*, 18: 280–315.

Měch, Radomír & Prusinkiewicz, Przemyslaw (1996): Visual models of plants interacting with their environment. *Computer Graphics Proceedings, Annual Conference Series*, 1996; SIGGRAPH 96, New Orleans, August 1996: 397–410.

Prusinkiewicz, Przemyslaw & Hanan, James (1989): Lindenmayer Systems, Fractals, and Plants. *Lecture Notes in Biomathematics*, 79, New York: Springer (120 pp.).

Prusinkiewicz, Przemyslaw & Lindenmayer, Aristid (1990): The Algorithmic Beauty of Plants. New York: Springer. (http://algorithmicbotany.org/papers/abop/abop.pdf).

Prusinkiewicz, Przemyslaw & James, Mark & Měch, Radomír (1994): Synthetic topiary. *Computer Graphics Proceedings, Annual Conference Series*, 1994; SIGGRAPH 94, Orlando, 24.-29. 7. 1994: 351–358.

Rozenberg, Grzegorz (1973): T0L systems and languages. *Information and Control*, 23: 357–381.

Rozenberg, Grzegorz (1997): *Handbook of Graph Grammars and Computing by Graph Transformations*. Vol. 1: Foundations. Singapore: World Scientific.

Vitanyi, Paul M. B. (1976): Digraphs associated with D0L systems. In: Lindenmayer, A. & Rozenberg, G. (eds.): *Automata, Languages, Development*. Amsterdam: North-Holland, pp. 335–346.

[Table of Content]

Tobias Isenberg

A Survey of Image-Morphologic Primitives in Non-Photorealistic Rendering

Abstract

This paper presents an overview of the image-morphologic primitives used commonly in non-photorealistic rendering (NPR), a subdomain of computer graphics that is inspired by a long tradition of artistic and illustrative depiction. In particular, we survey NPR shading, stroke-based rendering, sparse line drawings, graftals, and area primitives. Such primitives usually cover larger regions on the canvas and often carry a meaning beyond the color of the image region they represent. This distinguishes them from the pixel as a primitive used in photorealistic rendering, which does not have any meaning aside from sampling the color of the image section it represents. We give examples to illustrate the individual techniques and briefly mention how they are tracked though the rendering process as well as represented in the final image.

1 Introduction

For many years one main goal of computer graphics research has been to depict reality as it can be captured by photography. This goal of creating *photorealism* has received a lot of research attention in the areas of computer games and in the film industry. In the last decade or two, a different area of research has been established within computer graphics that does not share this goal. Instead, it is inspired by a long tradition of artistic and illustrative depiction and tries to break free from the constraints that are set by photorealistic rendering. This new area, *Non-Photorealistic Rendering* (NPR), has produced a wealth of techniques that allow us to simulate many forms of traditional media. For example, techniques such as oil painting, watercolor, pen-and-ink, stippling, or comic shading can now be reproduced fully automatically or using partial computer support; also entirely new techniques have been conceived. A good overview of the area give [Gooch & Gooch 2001] and [Strothotte & Schlechtweg 2002].

The new domain greatly benefits from the same freedoms that exist in traditional artistic and illustrative depiction such as the possibilities for abstraction, exaggeration, choice of view and projection, etc. Similarly, choices exist with respect to the selection of a tool and medium since NPR tries to emulate traditional means of depiction just as closely as photorealism tries to simulate the camera.[1] Tools and mediums in this context are, for example, brushes and watercolor for watercolor painting or copper plates and ink for pen-and-ink techniques. In a way one could also see optics as the particular and single tool of depiction in photorealism while in NPR there exists a variety of tools that can be chosen. On the one hand, this focus on tools for many NPR techniques results in creating marks that are evident in the produced images just like with traditional tools. On the other hand, these marks are also represented as primitives in the picture production process. Therefore, marks used in NPR can and usually do carry a meaning.

This constitutes a major shift from photorealistic rendering where images are rendered typically on a pixel-by-pixel basis. In photorealistic rendering, the triangles in a 3D model are traversed, and each triangle is rasterized into pixels, for which individual lighting and texturing computations are performed, and the pixel is finally stored into a buffer. In raytracing, the pixel is even more prevalent as rays are shot into the scene based on a predetermined pixel raster. This concentration on a pixel raster and the pixel as output primitive is somewhat arbitrary: the pixel raster is only determined by the overall pixel size of the output image and its resolution; it does not depend on the contents of the image itself. Pixels, therefore, do not carry a meaning beyond the color of the image section they represent.

In non-photorealistic rendering, in contrast, higher-level primitives are typically used to represent the depicted objects and scenes, even if the final image is rasterized and stored as a pixel matrix.[2] In contrast to pixels, higher-level primitives usually have a meaning beyond the essentially arbitrary measure of resolution. They normally represent the marks created by the traditional tools that are simulated in NPR. As such, NPR primitives can represent lighting conditions, similar to what pixels represent in photorealism. In contrast to photorealistic pixels, however, NPR primitives also represent properties of the depicted materials and objects. This is the case, for example, in hatching and stippling. Non-photorealistic rendering also allows us to go beyond the mere simulation of traditional techniques and make use of dynamic primitives such as *graftals* [Kowalski *et al*. 1999] that can adapt their way of rendering depending on conditions such as view and size on the screen.

We analyze the different types of image-morphologic primitives in non-photorealistic rendering in Section 2. We then explore how such primitives are typically tracked during the rendering process (Section 3), and how they are finally represented as either pixel images or vector graphics (Section 4). The final Section 5 concludes the paper.

---

[1] This could be seen as the quest for (photo)realism in NPR.

[2] This, however, does not have to be the case as NPR images can also be stored as vector graphics, depending on the rendering process.

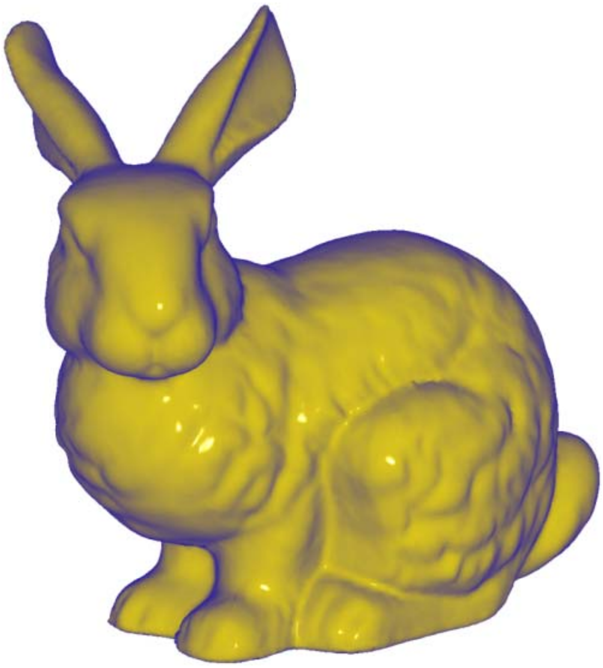


Figure 1: NPR cool-to-warm shading [Gooch et al*. 1998] can be used to better depict the surface shape. Cool colors (e.g., blue) are used in darker regions, warm colors (e.g., yellow) in lit parts*

2 Image-Morphologic Primitives in NPR

The different types of image-morphologic primitives that are being used in non-photorealistic rendering and imagery range from the pixel as in photorealistic rendering to fairly large elements such as silhouettes and feature strokes and even includes dynamic elements such as graftals. The following sections discuss these different types of primitives, grouped roughly by their size and their purpose in representing elements in the images.

2.1 Non-Photorealistic Shading

One subset of non-photorealistic rendering techniques adapts photorealistic rendering only slightly to create, for example, images that are more illustrative. This can be achieved, e.g., by using nonphotorealistic illumination models such as *Gooch cool-to-warm shading* [Gooch *et al*. 1998] (Figure 1). The goal of this technique is to introduce more richness into the transition from illuminated regions to dark regions, and to better suggest shape, a technique illustrators have been using for a long time. This is achieved by mixing the object's color properties with an additional transition from warm to cool colors, which makes color changes better visible in very bright regions as well as in very dark regions. As the traditional photorealistic rendering pipeline does not need to be changed much—only the illumination formula is modified—the same image-morphologic element is used: the pixel. As in photorealistic rendering, the pixel is employed as a means to sample the entire area of the image to be able to display and store it on digital media. Therefore, the pixels in

Figure 2: Application of Haeberli's [1990] Paint by Numbers technique to Figure 1 as the source image

these techniques are equivalent to pixels in photorealistic rendering: they do not carry any meaning aside from sampling the color of the image depending on the image size and resolution.

The vast majority of non-photorealistic rendering methods, however, are based on higher-level image-morphologic primitives such as strokes, mosaic tiles, graftals, area primitives, etc. This is partly the defined result of using abstraction, but mostly it is due to the goal of trying to simulate traditional techniques of artistic expression in which specific tools such as paintbrushes are used. These tools leave marks on the created images, giving the created images their unique character and style. Even though some techniques are implemented using pixel-by-pixel processing, these higher-level primitives are still evident in the produced images. One could argue that the tool in photorealistic rendering and non-photorealistic shading is the simulation of optical processes on a pixel-by-pixel basis resulting in no evidence of marks produced by this tool other than the pixels themselves. Therefore, the pixel as the primitive in non-photorealistic shading is the lowest-level primitive used in NPR.

2.2 Stroke-Based Rendering

A range of techniques that was very attractive for NPR researchers to attempt to replicate is the painting or drawing with strokes (e.g., brush strokes in oil and watercolor painting). The range of strokes being simulated includes, for example, pencil or ink strokes, stippling, and even the placement of decorative mosaic tiles. In their original use these marks can represent elements of the depicted scene (e.g., a brush stroke in painting could represent a leaf or a group of leaves in a tree) or just serve as a means to sample the scene for the canvas (e.g., pointillism). Marks in some techniques are used to work around limitations of the chosen medium. For example, stippling and hatching are employed due to the difficulty of using gray scales in the printing process.[3]

The area within NPR to simulate such techniques is called *stroke-based Rendering* [Hertzmann 2003]. The idea is to compute a new image based on an example image or 3D scene by rendering shorter or longer strokes. These strokes both approximate the example image as well as abstract from it at the same time. The degree of this approximation and abstraction depends on the specific types of marks, their size, and how many marks are being used.

Example techniques in this category include painterly rendering (e.g., [Meier 1996; Hertzmann 1998]), pointillism (e.g., [Yang & Yang 2006]), stippling (e.g., [Deussen *et al*. 2000; Secord 2002; Schlechtweg *et al*. 2005]), hatching (e.g., [Salisbury *et al*. 1994; Winkenbach & Salesin 1994; Salisbury *et al*. 1996; Winkenbach & Salesin 1996; Salisbury *et al*. 1997; Deussen *et al*. 1999; Ostromoukhov 1999; Hertzmann & Zorin 2000; Praun *et al.* 2001; Zander *et al*. 2004]), and the rendering of decorative mosaics (e.g., [Hausner 2001; Elber & Wolberg 2003; Di Blasi & Gallo 2005]).

One of the earliest and most influential approaches in stroke-based rendering was Haeberli's Paint by Numbers technique [1990].[4] His system introduced the principle of non-photorealistic abstraction of an image by placing strokes onto the target image using the color sampled from the source image. This approach did not yet have the target of closely simulating a specific traditional style, but used strokes as drawing primitives that are evident in the produced image (Figure 2), thus opening up a multitude of possibilities for stroke-based rendering in simulating traditional styles as well as coming up with new ones.

The evidence of mark making in traditional depiction techniques could be seen as an artifact of using the tool. However, artists and illustrators have developed ways to place marks that represent more than just that. This is very obvious, for example, in hatching as employed in woodcuts and copperplate engravings. Here, the hatching lines not only represent the gray value of an equivalent black-and-white photograph but also portray the structure and properties of the depicted surfaces. A masterly example for this application of marks is shown in Figure 3.

---

[3] Normally in the printing process, halftoning techniques are often employed which use dot patterns of pure colors (spot colors) to represent shades of gray or color.

[4] See http://laminadesign.com/explore/impression/imphelp.html for an example implementation of Haeberli's Paint by Numbers technique [1990].

Figure 3: Example off an artist's use of hatching to portray the structure and properties of the depicted surfaces (e.g., cloths and basket). Detail from the woodcut "Life of the Virgin: 14. The Rest during the Flight to Egypt" (1504–05) by Albrecht Dürer

NPR techniques simulating these pen-and-ink styles[5] have attempted to replicate this effect. For example, Salisbury *et al*. [1994] as well as Winkenbach & Salesin [1994] used specific *prioritized stroke textures* to represent different materials. These consist of prerecorded layers of strokes for representing a series of consecutively darker textures and are applied according to the gray value in a source image or the lighting conditions in a 3D scene. Other approaches put more emphasis on extracting a field of streamlines from a 3D surface to be able to illustrate the surface's shape (e.g., [Hertzmann & Zorin 2000] and [Zander *et al.* 2004]). These techniques portray illumination using line densities or by modulating the line parameters such as thickness or line stippling patterns (Figure 4).

In most pen-and-ink styles, strokes are used for showing the structure of surfaces by depicting ridges, creases, and other surface features (as done, for instance, by Winkenbach

---

[5] Although woodcuts and engravings are not technically pen-and-ink styles, NPR usually includes them in this group.

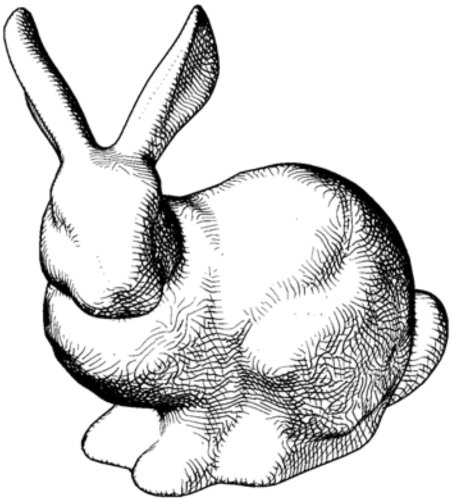


Figure 4: Hatching using *Zander* et al.*'s technique [2004]*

Figure 5: Painterly rendering: strokes represent leaves or wood shingles. Courtesy and copyright of Martin Schwarz, used with permission

and Salesin [1994]). In painterly rendering, on the other hand, strokes are typically employed to either represent whole elements of the depicted scene such as the leaves or wood shingles in Figure 5 or to convey the overall impression of painterly rendering.

In the latter case strokes are not associated with a particular object or scene element that they portray. In general, it can be difficult to algorithmically associate the placement of marks with elements of the depicted scene in a meaningful way as this assumes an understanding of the depicted scene. This can, however, be supported by allowing more interaction with the NPR rendering technique to guide the placement of marks in meaningful ways.

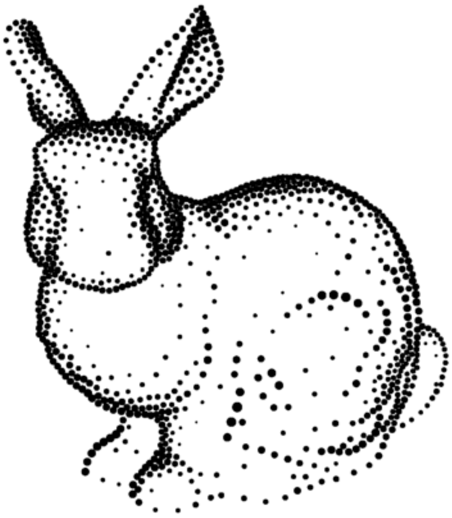


Figure 6: Stippling using Secord's technique [2002]

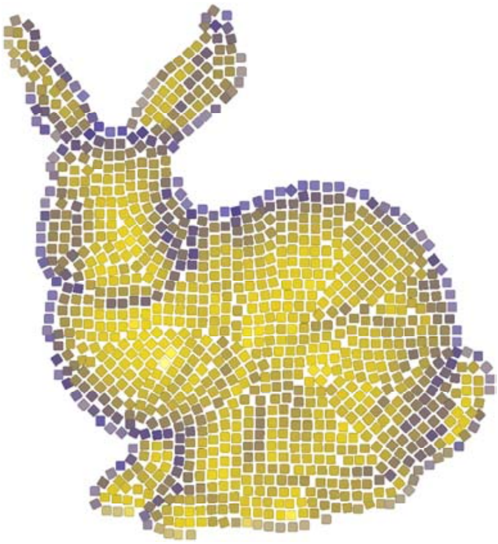
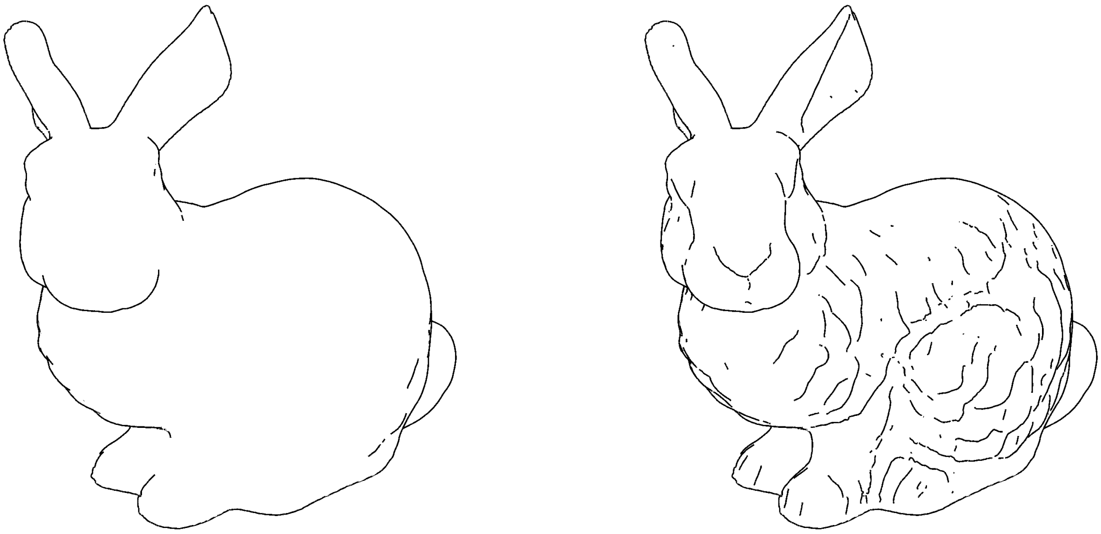


Figure 7: Decorative mosaics using Schlechtweg et al.*'s Render-Bots [2005]*



(a) Only silhouettes (b) With suggestive contours

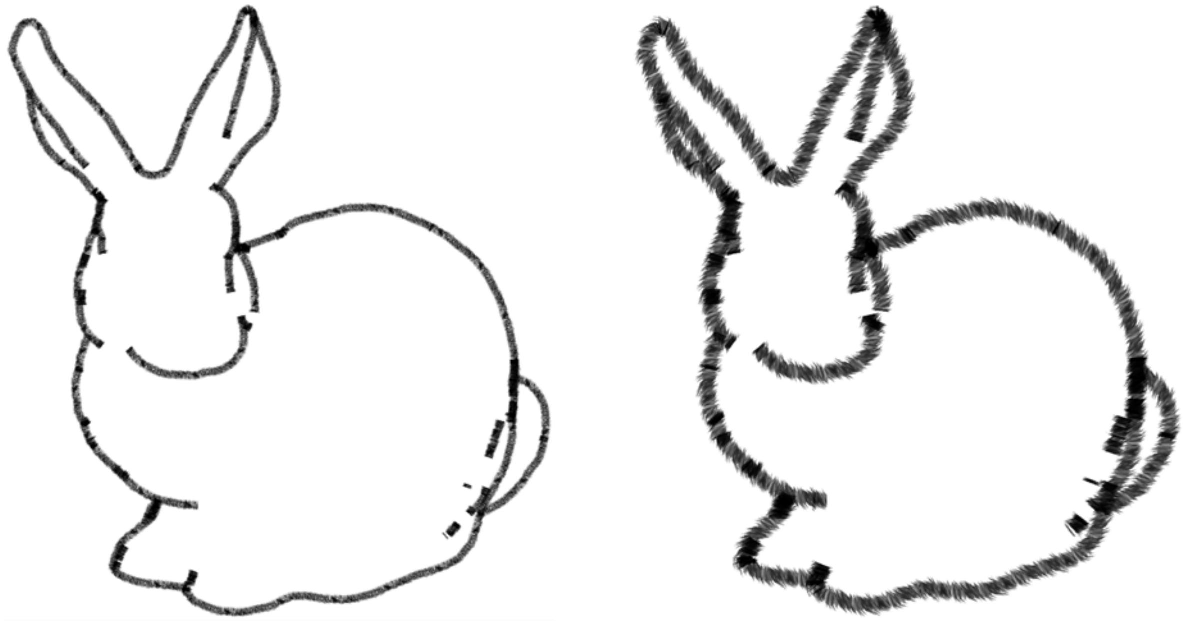
Figure 8: Comparison of a sparse line drawing with just silhouettes and with additional suggestive contours [DeCarlo et al. *2003]*

In some stroke-based rendering techniques such as stippling and (usually) the simulation of traditional mosaics, the primitives do not represent meaningful elements of the image. Instead, the marks are used to carry shading information (stippling, Figure 6) or colors (mosaic tiles, Figure 7) and are the—intended—artifacts of the specific technique.

2.3 Strokes in Sparse Line Drawings

Some meaningful structures can be algorithmically extracted from, in particular, 3D scenes and constitute exceptions to the above-mentioned general rule. These are *silhouettes* and *feature lines*; elements that make up *sparse line drawings* (abstract illustrations with just a few significant lines). Such lines are a very common means of expression and are traditionally used, for example, in comics and technical drawings. They are also often used in conjunction with the previously discussed stroke-based rendering techniques to guide stroke placement or as additional elements in the images.

Silhouettes are lines on the surface of 3D objects where the visibility changes from visible to invisible, or *vice versa* [Isenberg *et al.* 2003]. As such, silhouettes are view-dependent and move on the surface as a view onto the object changes. For closed objects, the silhouette also comprises a curve that borders the object and separates it from the background: its *contour*. Feature lines consist of lines on the surface that are otherwise significant and should be drawn in a sparse line drawing. The latter group includes view-independent creases, i.e., lines of sharp bends or high local curvature of the surface, but also view-dependent lines such as *suggestive contours* [DeCarlo *et al.* 2003] that visually extend the silhouettes in an image (Figure 8).



(a) Charcoal style (b) Fuzzy style

Figure 9: Applying line styles in form of textures to extracted silhouette strokes

After having been extracted from a 3D scene, silhouette and feature line segments are concatenated to form longer strokes. This character of forming long strokes solely as meaningful elements of the image distinguishes the lines in sparse line drawings from the strokes used in stroke-based rendering where strokes are also and probably mainly a means of sampling the image space, i.e., are used for portraying color and/or shading.

The silhouette or feature strokes can now either be drawn directly or be modified by applying a *line style*. The latter can simulate a specific traditional drawing tool such as a pencil, watercolor, chalk, or charcoal by applying an appropriate texture (Figure 9). Line styles can also disturb the path of a stroke, for example, to simulate the appearance of sketchiness.

2.4 Graftals

Graftals are a special form of stroke used in non-photorealistic rendering. Introduced to computer graphics by Smith [1984], in NPR the term has developed to comprise primitives that can algorithmically change their visual representation depending on parameters that can vary over the course of an animation or simulation [Kowalski *et al.* 1999; Kaplan *et al.* 2000; Markosian *et al.* 2000]. Similar to some elements in stroke-based rendering, they represent meaningful elements in a scene. In contrast to stroke-based rendering, however, the dynamic and procedural character allows graftals to change the visual representation depending on, e.g., the orientation or distance to the viewer (Figure 10). For example, a tuft of grass in the background may be shown with just one or two black strokes on a

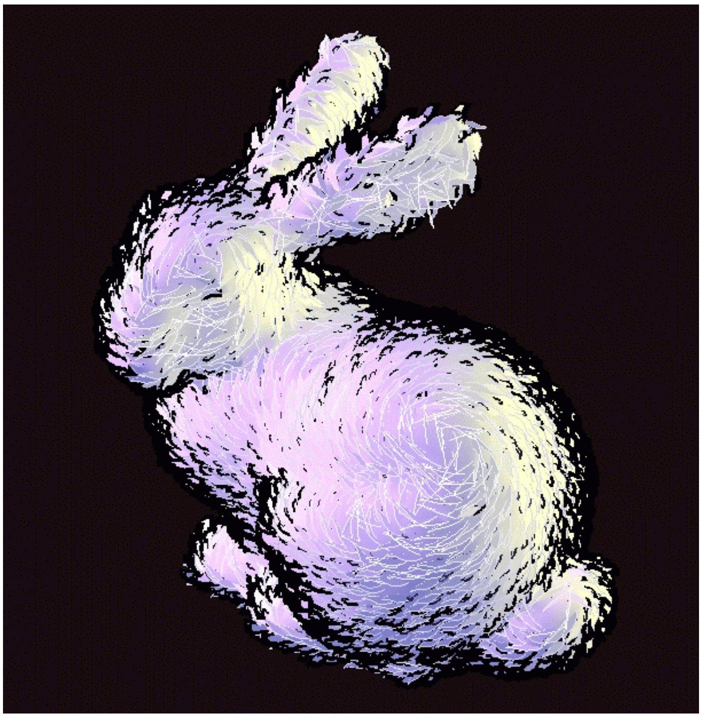


Figure 10: Graftals used to simulate artistic fur [Kaplan et al*. 2000]. Note the different appearance of the graftals facing the viewer and those on surface parts that are perpendicular to the viewing direction. Image courtesy and copyright of Matthew Kaplan, used with permission*

green background. As the camera gets closer, more strokes will appear and eventually a bush of triangular leaves will become visible.

Initially, graftals were used in an image-space stroke-based rendering manner [Kowalski *et al*. 1999]. This, however, leads to frame-incoherence and, thus, to flickering images in an animation because of the frame-by-frame processing and each frame being treated independently. In other approaches, graftals are placed into the scene during the modeling phase, directly associating them to the specific surfaces or objects they represent [Kaplan et al. 2000; Markosian *et al*. 2000]. This way their locations remained fixed on the respective surfaces, which makes the rendering coherent over time. In a way, this contrast between frame-incoherence vs. the maintenance of primitives over time constitutes a temporal equivalent of the difference between (very local) pixel processing and higher-level primitives.

2.5 Area Primitives and Patterns

One final group of primitives that are used in non-photorealistic rendering is not primarily based on short strokes, long strokes, or procedural graftals. We call this group *area primitives and patterns* because they cover larger areas of the produced images, sometimes filling them with patterns. NPR techniques that focus on the use of real primitives simulate, for example, traditional ornaments, modern art styles, and cell animation. As they do not

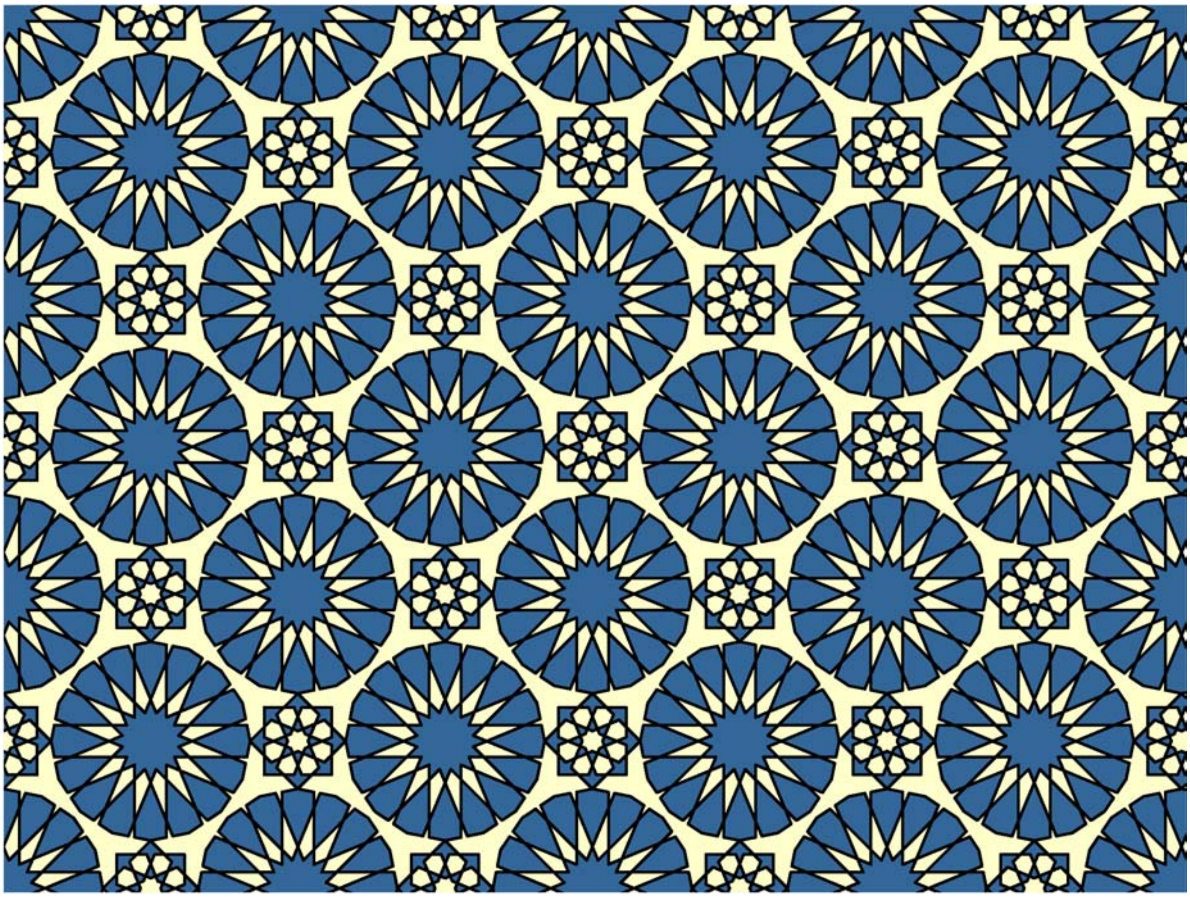


Figure 11: Oriental pattern generated using the method by Kaplan & Salesin [2004]

necessarily have a common algorithmic background, we restrict ourselves here to a few examples. Ornaments are algorithmically produced in form of floral, oriental, and Celtic patterns as inspired by the traditional examples [Ostromoukhov 1998; Wong *et al*. 1998; Kaplan & Cohen 2003; Kaplan & Salesin 2004], usually based on a mathematic scheme underlying the ornament (Figure 11). Related approaches reproduce ornamental effects such as the ones demonstrated by the works of M. C. Escher where the plane is entirely filled by repeating tiles of a given input image [Kaplan & Salesin 2000] or where two tiles are used which are morphed into one another [Ostromoukhov & Hersch 1995]. Other techniques are inspired by works by Piet Mondrian or the Japanese Seigaiha style and employ multi-agent systems and coalition forming to generate the elements in an image such as lines, colored tiles, and other patterns [Mason *et al.* 2005].

Cel shading can be thought of as a special case of area primitives. While it is technically an NPR shading technique (as discussed in Section 2.1), it also generates area features that can be regarded as non-photorealistic image-morphologic primitives. Here, the typical Phong shading technique of surfaces is changed such that regions with solid colors are created (Figure 12). This is inspired by traditional cel animation where foreground figures were drawn using silhouettes and feature lines on celluloid, and the regions then filled-in with color; finally the figure was recorded on a background. In NPR, the effect of a few

(a) Two colors (b) Three colors

Figure 12: Cel shading with two or three color steps, combined with silhouettes

shading levels (e.g., shadow, regular colors, and highlights) can be created by defining thresholds for the illumination of a surface, and then coloring all points that lie between two thresholds with the same color. There are also techniques to track highlights specifically, and to give them distinct shapes [Anjyo & Hiramitsu 2003].

3 Tracking Image-Morphologic Primitives

While the algorithms to create the primitives described above are as plentiful as there are primitives and NPR effects, there are generally three distinct categories they can be attributed to. These are image-space, hybrid, and object-space techniques. Depending on the category an algorithm belongs to, the generated primitives are explicitly represented during the rendering process or not. Even if primitives are not explicitly represented in the rendering process they may still be present in the final image.

We briefly discuss the three groups here using the example of silhouette and feature line extraction (for more detail see [Isenberg *et al*. 2003]). Image-space or pixel-based silhouette extraction depends on additional G-buffers[6] storing depth (z-buffer) and/or normal vector[7] information. This data can be processed using an edge detection filter [Saito & Takahashi 1990], resulting in purely local edge elements being detected where discontinuities of the depths or surface orientations are in these G-buffers, i.e., at silhouettes and fea-

---

[6] G-buffers are pixel images of the same size as the rendered image. They are generated during the rendering process and store geometric information about the rendered scene such as local depth values, normal vectors, ID values of triangles or objects, etc.

[7] The normal vector of a point on the surface of an object is perpendicular to the tangential plane that touches the surface point.

ture lines. Thus, even though during the process no silhouette or feature line is explicitly represented, they are nevertheless created through the local pixel processing.

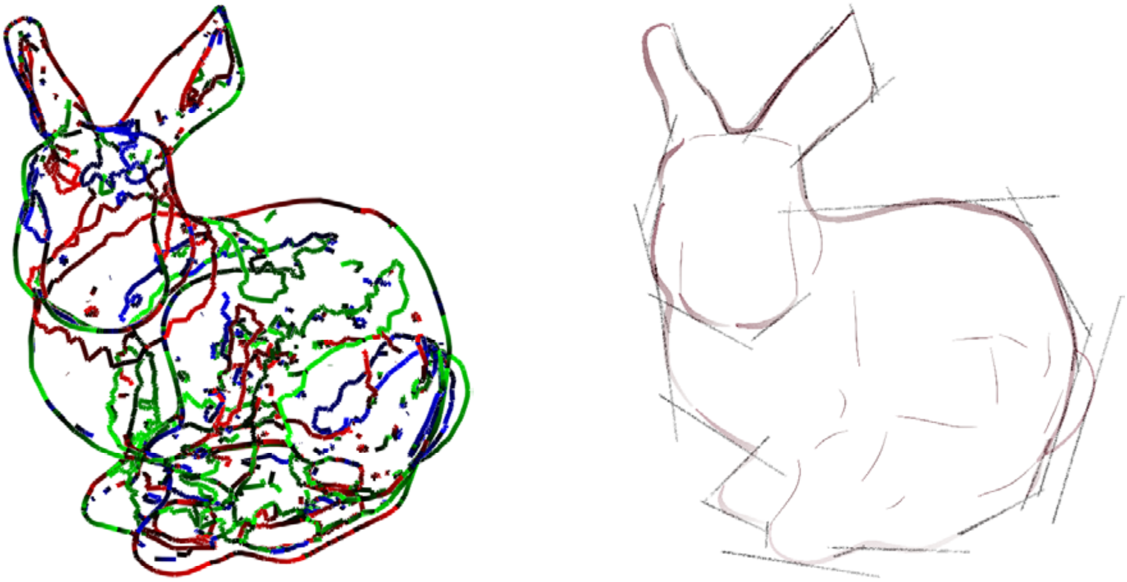
Hybrid techniques do not rely on a pixel-by-pixel processing directly but arrange the actual rendering in a smart way so that silhouettes are created. For example, they select polygons on the backside of an object first, enlarge them, and render them in black. Afterwards, the polygons on the front side are rendered normally; but at the silhouettes the previously rendered black polygons stick out, forming silhouette lines. Again, the silhouettes are not explicitly represented but exist in the final image.

In contrast, object-space techniques do explicitly extract and represent the intended primitives in the rendering process. In the case of silhouette extraction, visibility information is determined, for example, on a polygonal mesh, and line segments are then identified where this visibility changes (from facing a viewer/camera to facing away, or *vice versa*). These line segments are concatenated to form long strokes based on connectivity information from the mesh and can then be rendered.

As a second example, the cel shading technique discussed in Section 2.5 could be thought of as an image-space technique (because it applies a threshold to the computed pixel colors) or a hybrid method (because it changes the lighting computation during rendering). Nevertheless, the image-morphologic primitives generated by this technique—regions with a constant color—only become apparent in the final image and are not present during the rendering process. However, cel shading may be also created in an object-space manner as recently demonstrated by Stroila *et al.* [2008]. Here, the curves bordering the regions of uniform color are explicitly extracted from the scene, and shapes representing the regions are created from these borders.

The characteristic of primitives that they are explicitly represented in the rendering process is essential where the primitives have to undergo further processing. Primitives that are only present in the resulting image in form of pixel colors usually cannot be altered on a meaningful primitive-by-primitive level; only methods such as image-processing filters that work on a pixel-by-pixel basis can be applied. Therefore, in particular in domains such as sparse line drawings where image elements need to be stylized, object-space techniques are used to extract the primitives. Several methods have been created to aid this stylization process through capturing and maintaining additional properties needed for it.

Grabli *et al.* [2004] capture information such as extracted strokes, their type, their visibility, and a number of other data items in a graph data structure called *view map* (Figure 13(a)) that lets them algorithmically determine which strokes to select, chain, and stylize in a wide variety of ways. These complex styles can be stored in style sheets in order to enable easy re-using. This approach allows them to create more complex and elaborate stylized



(a) Color-coded viewmap (b) Example drawing

Figure 13: Images generated with the system by Grabli et al. *[2004]*

line drawings as still images (Figure 13(b)). Isenberg & Brennecke [2006] introduced their *G-strokes* approach, which also captures stroke properties but stores them as information tracks parallel to the stroke's geometry data. As the geometric stroke data is processed in the rendering pipeline, so is the additional G-stroke data. For example, when a stroke's visibility is determined, a segment may be found that needs to be split because the visibility changes along its path. In that case an additional visibility G-stroke can be used to capture the visibility information. This new G-stroke and all others G-strokes are adapted during rendering to reflect the necessary changes. In the case of the visibility change this requires splitting the geometry of the segment as well as all its other G-stroke data. This way it is possible to create complex stroke pipeline networks to stylize sparse line drawing at interactive rates (Figure 14).

4 Image Representation

The distinction of whether NPR rendering primitives are explicitly represented in the rendering process or not also plays a role in how the produced images are output and stored. This can occur in one of two forms: as *pixel images* or as *vector graphics*. The type of image determines whether primitives can be explicitly represented in the image as well.

Most commonly, images are stored in pixel raster form. This means that for a given size and resolution a raster is mapped onto the image, and each pixel samples a color, which is then stored in the image. The rendering process itself may already determine this raster, for instance when shading techniques are used. Because of the pre-defined sampling on the pixel raster, and as already noted in Sections 1 and 2.1, pixel images do not represent NPR primitives explicitly. Therefore, it is difficult to identify them algorithmically. However,

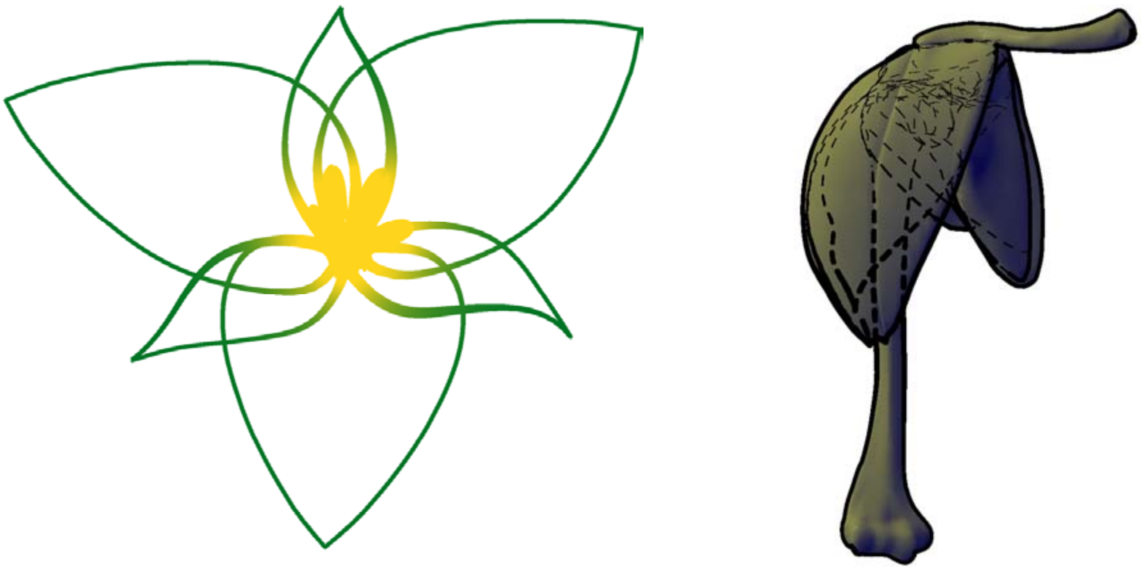


Figure 14: Using G-Strokes by Isenberg & Brennecke [2006] in conjunction with NPR Lenses [Neumann et al*. 2007] to apply local style changes to line drawings*

pixel images are the by far most often used form of image representation and are supported by virtually all systems where images may be needed.

The second class of image representations, vector graphics, does represent primitives as separate structures in the stored files. Therefore, it is easier to maintain primitives as separate entities in the final image as well and to allow further processing (e.g., changing stroke paths or selecting subsets of primitives). Vector images also do not have an inherent resolution and are rasterized to the resolution needed for a specific case which in most cases leads to a higher quality in the representation. It also results in their data volume usually being smaller than the equivalent pixel image, depending on the number and complexity of elements stored and the resolution of the pixel image [Isenberg *et al*. 2005]. However, as vector images store elements in analytic form, they also have to be interpreted every time they are displayed, thus requiring more time for this process than pixel images.

5 Conclusion

In this paper we have compared groups of image-morphologic primitives in non-photorealistic rendering. These include NPR shading techniques, stroke-based rendering, the generation of sparse line drawing, graphtals, and area primitives. We have shown that NPR techniques tend to work with elements that are larger than an individual pixel, usually inspired by the tools used traditionally in artistic depiction and the marks created by them. We have discussed that some techniques add such artifacts to images to give the impression of the traditional technique but that there are also a number of techniques where marks actually correlate to meaningful structures in images. We showed that the primitives may or may not be explicitly represented in the rendering process, resulting in more or less freedom for subsequently changing the appearance of the elements. Finally, we briefly touched on that this explicit representation might be carried over to the stored image in form of a vector graphic, which allows higher quality reproduction as well as post-processing on the primitive level.

The overview that is given in this paper presents a morphologic (i.e., syntactic) view of the primitives used in non-photorealistic rendering. This naturally does not touch on the important issues of semantics and, in particular, pragmatics of the images produced in such processes.

Acknowledgments

We would like to thank Alberta Ingenuity (AI) for funding this research. We also thank Sheelagh Carpendale, Pauline Jepp, Petra Neumann, and Martin Schwarz for fruitful discussions on the topic.

References

Anjyo, K.-I. & Hiramitsu, K.: Stylized Highlights for Cartoon Rendering and Animation. IEEE Computer Graphics and Applications 23, 4 (July/Aug.), 2003, 54–61.

DeCarlo, D. & Finkelstein, A. & Rusinkiewicz, S. & Santella, A.: Suggestive Contours for Conveying Shape. ACM Transactions on Graphics 22, 3 (July), 2003, 848–855.

Deussen, O. & Hamel, J. & Raab, A. & Schlechtweg, S. & Strothotte, T.: An Illustration Technique Using Hardware-Based Intersections and Skeletons. In Proc. of GI'99, Morgan Kaufmann, San Francisco, 1999, 175–182.

Deussen, O. & Hiller, S. & Van Overveld, C. W. A. M. & Strothotte, T.: Floating Points: A Method for Computing Stipple Drawings. Computer Graphics Forum 19, 3 (Aug.), 2000, 40–51.

Di Blasi, G. & Gallo, G.: Artificial Mosaics. The Visual Computer 21, 6 (July), 2000, 373–383.

Elber, G. & Wolberg, G.: Rendering Traditional Mosaics. The Visual Computer 19, 1 (Mar.), 2003, 67–78.

Gooch, B. & Gooch, A. A.: Non-Photorealistic Rendering. A K Peters, Natick, 2001.

Gooch, A. A. & Gooch, B. & Shirley, P. & Cohen, E.: A Non-Photorealistic Lighting Model for Automatic Technical Illustration. In Proc. of ACM SIGGRAPH'98, ACM Press, New York, 1998, 447–452.

Grabli, S. & Turquin, E. & Durand, F. & Sillion, F. X.: Programmable Style for NPR Line Drawing. In Rendering Techniques 2004, Proc. of the EG Symp. on Rendering'04, EUROGRAPHICS Association, Aire-la-Ville, Switzerland, 2004, 33–44, 407.

Haeberli, P.: Paint By Numbers: Abstract Image Representations. ACM SIGGRAPH Computer Graphics 24, 3 (Aug.), 1990, 207–214.

Hausner, A.: Simulating Decorative Mosaics. In Proc. of ACM SIGGRAPH'01, ACM Press, New York, 2001, 573–580.

Hertzmann, A. & Zorin, D.: Illustrating Smooth Surfaces. In Proc. of ACM SIGGRAPH'00, ACM Press, New York, 2000, 517–526.

Hertzmann, A.: Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In Proc. of ACM SIGGRAPH’98, ACM Press, New York, 1998, 453–460.

Hertzmann, A.: A Survey of Stroke-Based Rendering. IEEE Computer Graphics and Applications 23, 4 (July/Aug.), 2003, 70–81.

Isenberg, T. & Brennecke, A.: G-Strokes: A Concept for Simplifying Line Stylization. Computers & Graphics 30, 5 (Oct.), 2006, 754–766.

Isenberg, T. & Carpendale, M. S. T. & Sousa, M. C.: Breaking the Pixel Barrier. In Proc. of CAe’05, Eurographics Association, Aire-la-Ville, Switzerland, 2005, 41–48.

Isenberg, T. & Freudenberg, B. & Halper, N. & Schlechtweg, S. & Strothotte, T.: A Developer’s Guide to Silhouette Algorithms for Polygonal Models. IEEE Computer Graphics and Application 23, 4 (July/Aug.), 2003, 28–37.

Kaplan, M. & Cohen, E.: Computer Generated Celtic Design. In Rendering Techniques’03, Eurographics Association, Aire-la-Ville, Switzerland, 2003, 9–19.

Kaplan, C. S. & Salesin, D. H.: Escherization. In Proc. of ACM SIGGRAPH’00, ACM Press, New York, 2000, 499–510.

Kaplan, C. S. & Salesin, D. H.: Islamic Star Patterns in Absolute Geometry. ACM Transactions on Graphics 23, 2 (Apr.), 2004, 97–119.

Kaplan, M. & Gooch, B. & Cohen, E.: Interactive Artistic Rendering. In Proc. of NPAR’00, ACM Press, New York, 2000, 67–74.

Kowalski, M. A. & Markosian, L. & Northrup, J. D. & Bourdev, L. & Barzel, R. & Holden, L. S. & Hughes, J. F.: Art-Based Rendering of Fur, Grass, and Trees. In Proc. of ACM SIGGRAPH’99, ACM Press, New York, 1999, 433–438.

Markosian, L., Meier & B. J., Kowalski, M. A. & Holden, L. S. & Northrup, J. D. & Hughes, J. F.: Art-based Rendering with Continuous Levels of Detail. In Proc. of NPAR’00, ACM Press, New York, 2000, 59–64.

Mason, K. & Denzinger, J. & Carpendale, S.: Negotiating Gestalt: Artistic Expression by Coalition Formation between Agents. In Proc. of Smart Graphics’05, Springer, Berlin, vol. 3638 of LNCS 2005, 103–114.

Meier, B. J.: Painterly Rendering for Animation. In Proc. of ACM SIGGRAPH’96, ACM Press, New York, 1996, 477–484.

Neumann, P. & Isenberg, T. & Carpendale, S.: NPR Lenses: Interactive Tools for Non-Photorealistic Line Drawings. In Proc. of Smart Graphics’07, Springer, Berlin, Heidelberg, vol. 4569 of LNCS, 2007, 10–22.

Ostromoukhov, V. & Hersch, R. D.: Artistic Screening. In Proc. of ACM SIGGRAPH’95, ACM Press, New York, 1995, 219–228.

Ostromoukhov, V.: Mathematical Tools for Computer-Generated Ornamental Patterns. In Electronic Publishing, Artistic Imaging and Digital Typography. Springer, Berlin, 1998, 193–223.

Ostromoukhov, V.: Digital Facial Engraving. In Proc. of ACM SIGGRAPH’99, ACM Press, New York, 1999, 417–424.

Praun, E. & Hoppe, H. & Webb, M. & Finkelstein, A.: Real-Time Hatching. In Proc. of ACM SIGGRAPH’01, ACM Press, New York, 2001, 581–586.

Saito, T. & Takahashi, T.: Comprehensible Rendering of 3-D Shapes. ACM SIGGRAPH Computer Graphics 24, 3 (Aug.), 1990, 197–206.

Salisbury, M. P. & Anderson, S. E. & Barzel, R. & Salesin, D. H.: Interactive Pen-and-Ink Illustration. In Proc. of ACM SIGGRAPH’94, ACM Press, New York, 1994, 101–108.

Salisbury, M. P. & Anderson, C. & Lischinski, D. & Salisin, D. H.: Scale-Dependent Reproduction of Pen-and-Ink Illustration. In Proc. of ACM SIGGRAPH’96, ACM Press, New York, 1996, 461–468.

Salisbury, M. P. & Wong, M. T. & Hughes, J. F. & Salesin, D. H.: Orientable Textures for Image-Based Pen-and-Ink Illustration. In Proc. of ACM SIGGRAPH’97, ACM Press, New York, 1997, 401–406.

Schlechtweg, S. & Germer, T. & Strothotte, T.: RenderBots—Multi Agent Systems for Direct Image Generation. Computer Graphics Forum 24, 2 (June), 2005, 137–148.

Secord, A.: Weighted Voronoi Stippling. In Proc. of NPAR’02, ACM Press, New York, 2002, 37–44.

Smith, A. R.: Plants, Fractals, and Formal Languages. ACM SIGGRAPH Computer Graphics 18, 3 (July), 1984, 1–10.

Stroila, M. & Eisemann, E. & Hart, J. C.: Clip Art Rendering of Smooth Isosurfaces. IEEE Transactions on Visualization and Computer Graphics. 14, 1 (Jan./Feb.), 2008, 135--145.

Strothotte, T. & Schlechtweg, S.: Non-Photorealistic Computer Graphics. Modeling, Animation, and Rendering. Morgan Kaufmann, San Francisco, 2002.

Winkenbach, G. & Salesin, D. H.: Computer-Generated Pen-and-Ink Illustration. In Proc. of ACM SIGGRAPH’94, ACM Press,New York, 1994, 91–100.

Winkenbach, G. & Salesin, D. H.: Rendering Parametric Surfaces in Pen and Ink. In Proc. of ACM SIGGRAPH’96, ACM Press, New York, 1996, 469–476.

Wong, M. T. & Zongker, D. E. & Salesin, D. H.: Computer-Generated Floral Ornament. In Proc. of ACM SIGGRAPH’98, ACM Press, New York, 1998, 423–434.

Yang, H.-L. & Yang, C.-K.: A Non-Photorealistic Rendering of Seurat’s Pointillism. In Advances in Visual Computing, Part 2, Springer, Berlin, 2006, 760–769.

Zander, J. & Isenberg, T. & Schlechtweg, S. & Strothotte, T.: High Quality Hatching. Computer Graphics Forum 23, 3 (Sept.), 2004, 421–430.

[Table of Content]

Hans Du Buf, Joao Rodrigues

Image Morphology: From Perception to Rendering

Abstract

A complete image ontology can be obtained by formalising a top-down meta-language which must address all possibilities, from global message and composition to objects and local surface properties. In computer vision, where one general goal is image understanding, one starts with a bunch of pixels. The latter is a typical example of bottom-up processing, from pixels to objects to layout and gist. Both top-down and bottom-up approaches are possible, but can these be unified? As it turns out, the answer is yes, because our visual system does it all the time. This follows from our progress in developing models of the visual system, and using the models in re-creating an input image in the form of a painting.

1 Introduction

A trained painter is able to look at a scene and almost instantaneously take decisions concerning composition (spatial and semantic relations between objects), abstraction (which objects to paint and the level of detail) and techniques (colour palette, brushes and stroke types). Painting can be done very fast (wet-in-wet) when mixing colours on the canvas, or in different sessions (wet-on-dry) for applying new layers. Most painters who apply traditional styles will work from background to foreground, even with the possibility to start the background with dark colours and finish by high-lighting important regions using bright colours (clair-obscur or chiaroscuro). If it were possible to take a look into the brain of painters and unravel all the processes that are going on, we could develop a sound theory. We wrote “could” instead of “can” because of complications that can be expected: every painter has developed an own style, and it is likely that a specific style is related to a specific way the “input image” has been or is being analysed.

Unfortunately, we cannot take a look into van Gogh’s head and we do not know the exact landscapes that he saw. We can only analyse the paintings that he produced. Functional magnetic-resonance imaging (fMRI), which is a relatively new technology for analysing activities in brain areas, is not yet mature enough to be applied systematically. Besides, current fMRI technology lacks the resolution to analyse brain activity down to the cell level, i.e., only bigger regions and pathways between regions can be obtained. For the moment there are different and complementary solutions: (a) study composition and abstraction using methods employed in empirical aesthetics, (b) study specific visual effects such as colour and brightness using psychophysics, and (c) study available data concerning cells, layers and pathways using neurophysiology, hoping that basic processes in the brains of humans and other primates are the same or at least similar.

Here we will concentrate on visual perception and the visual cortex, without going too much into detail. One of the goals of the Vision Laboratory is to develop models of the visual cortex for explaining brightness effects and illusions, now also object categorisation and recognition. A new development is to apply low-level processing to non-photorealistic rendering (NPR), i.e., painterly rendering using discrete brush strokes. This combines two developments: a standard observer and a standard painter, with a user interface that allows to select e.g., brush and stroke types for influencing the painting process and therefore the style of the painting.

Below we first present a general description of the visual system and specific processes, including layers, pathways and cells, in the cortex. Then we illustrate how the cortical image representation can be used for NPR. We conclude with a Discussion in which we return to image ontology.

2 The visual system

The goal of our visual system, but in combination with the other senses, is to recognise objects, to establish a spatial layout of our environment, and to prepare for actions, for example looking at a computer monitor and keyboard when typing a text. All this is done automatically and very fast. In addition, the image that we perceive looks perfect for those without deficiencies—except for vision scientists familiar with illusions. However, how all this is done is still a mystery. Despite the tremendous progress in research during the past decades, there still remain many open questions although our view of basic processes has become clearer. A few aspects are the following:

2.1 The retina

The projected image on the retina is pre-processed there: rods and cones, the basic photoreceptors, are connected by horizontal cells with excitative and inhibitory synapses, a first indication for spatial (or spatio-temporal) filtering. They are also connected to bipolar cells which connect to amacrine and ganglion cells. Already 12 types of bipolar cells have been identified, with at least 4 types of ON and OFF cone-connected cells. Cones play a role in daylight colour vision whereas rods are for black-white vision when the light level is low. ON and OFF refer to light increments and decrements on a background, for example white and black spots or bars on a grey background. Amacrine cells are inhibitory interneurons of ganglion cells, and as many as 50 morphological types exist. At least 10–15 types of retinal ganglion cells have been identified. These code ON and OFF signals for spatial, temporal, brightness and colour processing, and their outputs, the axons, connect to the lateral geniculate nucleus (LGN) and other brain areas (the LGN is a relay station between the retina and the visual cortex, input area V1; see below). For further details we refer to Wässle (2004).

Most important here is that receptive fields of ON and OFF retinal ganglion cells can be seen as isotropic spatial bandpass filters, i.e., without a preferred orientation and therefore with a circularly-symmetric point spread function, often modelled by means of a “Mexican hat” function with a positive centre and a negative surround. Such filters only respond to transitions like dark-bright edges, and responses in homogeneous regions are zero or very small. The size of the receptive fields is a function of the retinal eccentricity: the fields are small in the centre (fovea) and they are increasingly bigger towards the periphery. According to another theory (!), big fields exist over the entire retina, medium fields inside a circular region around the fovea, and the smallest fields are only found in the centre of the fovea. Related to the field size is the notion of scale representation: at the point that we fixate fine-scale information is available, for example for resolving printed characters of a text we are reading, whereas the surround is blurred because only medium-and coarse-scale information is available there. The notion of scale analysis or scale representation will become clearer in Section 3.

Also important is the fact that one very specific type of retinal ganglion cell is not connected, directly nor indirectly, to rods and cones (Berson 2003); their own dendrites act as photoreceptors, they have very big receptive fields, and they connect to central brain areas for controlling the circadian clock (day-night rhythm) and, via a feedback loop, the eye’s iris (pupil size). These special cells also connect to at least the ventral area of the LGN (LGNv); hence, in principle they can play a role in brightness perception, for obtaining a global background brightness on which lines and edges etc. are projected. This is still speculative and far from trivial, but we need to keep in mind that (a) pure bandpass filters, both retinal ganglion cells and cortical simple cells (see below), cannot convey a global (lowpass) background brightness level, (b) colour information is related to brightness and processed in the cytochrome-oxidase (CO) blobs embedded in the cortical hypercolumns, colour being more related to homogeneous image (object) regions instead of to lines and edges extracted on the basis of simple cells etc. in the hypercolumns and not in the CO blobs, (c) colour constancy, an effect that leads to the same perception of object colours when the colour of the light source (illumination spectrum) changes, is intrinsically related to brightness, i.e., in a more global sense rather than object edges etc., and (d) very fine dot patterns, for example a random pattern composed of tiny black dots on a white kitchen table, are difficult to code with normal retinal ganglion cells or cortical simple cells (Zucker

& Hummel, 1986; Allman & Zucker, 1990). Colour and dot-pattern processing suggest that there are more “pathways” from the retina to the visual cortex, although the availability of a cone-sampled image in the cortex is speculative (blindsight, the ability of a blind person to sense the presence of a light source or even a moving object, points at pathways that do not lead, at least directly, to area V1 in the cortex). Most of these aspects are subject to research. An amazing fact is that, in each eye, the information of 125 million rods and cones is coded by means of about one million retinal ganglion cells. The compression rate of 0.8% is impossible to achieve by current image and video compression standards like JPEG and MPEG if image quality may not deteriorate.

2.2 The LGN

The traditional view of the LGN is a passive relay station between the retina and V1, the cortical input layer that connects to higher areas V2, V4 etc. The more recent view is that the LGN plays an active role in visual attention: perhaps only 10% of its input stems from the retina and all other input it receives by means of feedback loops from inferior-temporal (IT) and prefrontal (PF) cortex, where short-term memory is thought to reside, via V4, V2 and V1. This implies that the magno and parvo subsystems, also called the ‘what’ and ‘where’ systems or pathways in ventral and dorsal areas throughout the visual cortex, already exist at LGN level: LGNv and LGNd (Kastner *et al.* 2006). The names ‘what’ and ‘where’ stem from the functionality of the system in testing hypotheses in the interpretation of the coded input information, i.e., what there is (object categorisation and recognition) and where it is (Focus-of-Attention and eye fixations). However, it should be stressed that the LGN is not involved in object recognition. Feedback from the visual cortex only modulates information passing through the LGN.

2.3 The visual cortex

The ‘what’ and ‘where’ pathways lead to V1 and via V2 and V4 to higher areas IT and PP (posterior-parietal). In the computational model by Deco and Rolls (2004), information in the ventral ‘what’ system propagates, bottom-up, from V1 via V2 and V4 to IT cortex. The dorsal ‘where’ system connects V1 and V2 through MT (medial-temporal) to PP. Both systems are controlled, top-down, by attention and short-term memory with object representations in PF cortex, i.e., a ‘what’ component from PF46v to IT and a ‘where’ component from PF46d to PP. Deco and Rolls showed that the bottom-up (visual input code) and top-down (expected object and position) data streams are necessary for obtaining size, rotation and translation invariance in object detection and recognition: object templates in memory are thought to represent a few canonical object views, probably normalised (if we close our eyes and imagine a few objects like a cup, a bottle, a cat and a house, one after the other, they all have more or less the same size). Invariance is obtained by dynamic routing in V2 and V4 etc., such that cells at higher levels (a) have bigger receptive fields until they cover the entire visual field, (b) perform more complex tasks, for example a face detector at a high level can combine outputs of eye and mouth detectors at a lower level, the eye and mouth detectors combining feature detectors at yet lower levels, and (c) can control attention and adapt/optimise local detection processes at the lower levels. Although Deco and Rolls (2004) explored attention and invariance, they did not apply any functional feature extractions, i.e., they only used simple cells in V1 instead of line, edge, keypoint and grating cells (see Section 3, which focuses on processing in area V1). A nice example of feature extraction is the multi-scale keypoint representation in V1 and beyond for face detection: the use of keypoints (singularities like line and edge crossings and end points) for detecting eyes etc. until a face is detected, see Rodrigues & du Buf (2006c). Such a hierarchical architecture can explain the well-known Thatcher illusion: the vertically mirrored picture with normal mouth and eye regions looks fine but when it is rotated it looks terrible. Explanation: mouth and eye detectors have no problem with the friendly facial expression and a face detector groups outputs of mouth and eye detectors; the mouth can be above or beneath the eyes, for the face detector this is the same when it only groups outputs of the other detectors.

2.4 Information propagation

Although we can detect and recognise objects very fast, almost instantaneously as it seems, processing in the different cortical areas and the information propagation, both bottom-up and top-down, take time. When seeing an image for a split second, we are able to extract the gist and detect specific objects. What happens is that the flashed image enters the system and, after the computer screen goes blank again, the information propagates through the different levels (the same occurs between fixations, during saccadic eye movements when the image is not stable and the input is inhibited). Typically, objects are recognised within 150–200 ms, and first category-specific activation of PF cortex starts after about 100 ms (Bar 2004). In addition, instead of all information propagating at the same time, or in parallel, it is known that coarse-scale information propagates faster than fine-scale information to IT cortex (Bar *et al.* 2006). This suggests that object segregation, categorisation and recognition are sequential but probably overlapping processes: the system starts with coarse scales for a first test to select possible object templates, then employs medium scales in order to refine the categorisation, until finest scales are available for final confirmation of the recognition result. For another view of the cortical architecture we refer to Rensink (2000). Rensink explains the fact that the “bandwidth” of the visual system is limited: only one object can be attended at any time, although the presence of multiple objects must be stored in what he calls layout and gist subsystems. He also explains that our brain does not need to store a complete map of our entire environment; the (normally) stable environment we are looking at can be seen as external memory. Indeed, when we close our eyes we are very poor in naming colours and other aspects of objects

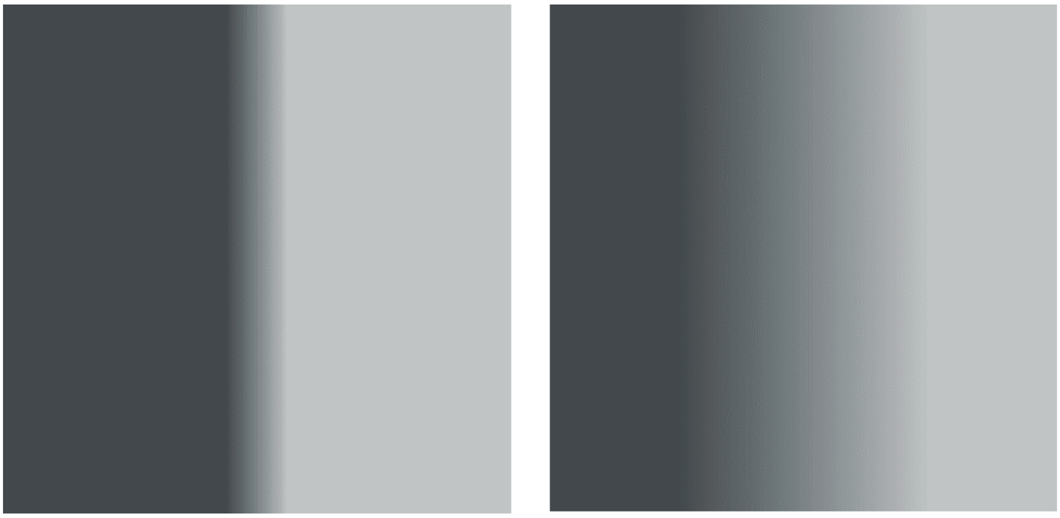


Figure 1: Both narrow and broad but linear transitions between dark and bright regions lead to the perception of Mach bands, a dark band to the left and a bright one to the right in both images. This illusion is explained in the text and in Figure 4

that are on the table in front of us. In vision science a related effect is called change blindness: when looking sequentially at two images of a house, the one with a chimney to the left and the other with the same chimney but moved to the right, only few people will notice the difference. Apparently, the house looks normal (gist), the position of the chimney is irrelevant (layout), and the system can spend its limited “bandwidth” on more important tasks, until we are told to look for differences and we start screening consciously different parts of the two images.

Above we did not address other issues like motion and disparity. In the next section we will focus on feature extractions in V1, by means of specialised cells. But some general questions remain: if things are quite complicated, with still many gaps in our knowledge, how is the image created that we perceive? Where in our brain is it created? Well, nobody knows exactly, but researchers who are developing, e.g., computational brightness models should have an idea. If we require that a brightness model should at least be able to predict Mach bands, the bright and dark bands that are seen at ramp edges (see Fig. 1), the number of published models is surprisingly small (Pessoa 1996).

If, in addition, we require that a model that can predict Mach bands should also be able to predict most of all known brightness illusions like brightness induction, with the two opposite effects of simultaneous brightness contrast and assimilation (see Fig. 2), the number of models is even smaller. Our own model was first tested on 1D patterns (du Buf 1994; du Buf & Fischer 1995), but a 2D version has already been tested and will soon be submitted for publication. It is based on a specific philosophy that answers the two questions posed above.

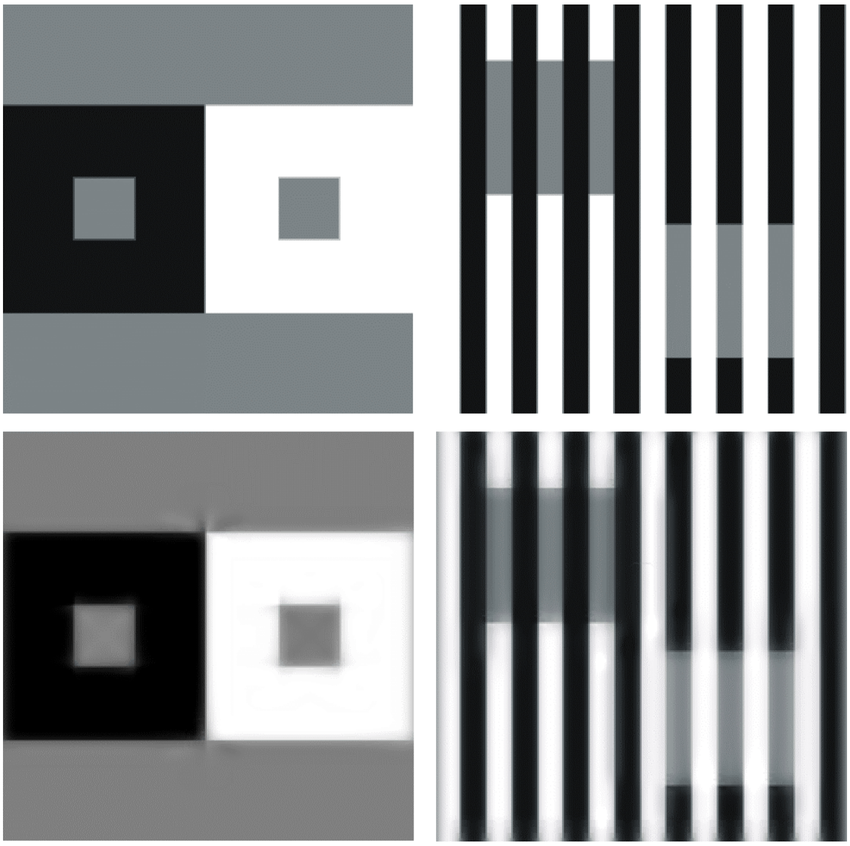


Figure 2: Top: two examples of brightness induction, simultaneous brightness contrast (left) and assimilation (right). In both images the grey squares and the bars are of the same intensity physically, but there is a big difference in our brightness interpretation; Bottom: model predictions show correct effects

3 Feature extractions in V1 and brightness perception

V1 is the input layer of the visual cortex in both left and right hemispheres of the brain. It is organised in so-called cortical hypercolumns, with neighbouring left-right regions which receive input—via the optic chiasm and one of the two LGNs—from the left and right eyes, with small “islands,” the CO blobs. In the hypercolumns there are simple, complex and end-stopped cells. Simple and complex cells are thought to serve line and edge extraction, whereas end-stopped cells respond to singularities (line/edge crossings, vertices, end points). There are many cells tuned to different scales, i.e., with receptive fields that range from very small to very big. If we penetrate the surface of the cortex perpendicularly, we find cells tuned to different orientations. Many cells are also disparity-tuned, which indicates that stereo processing starts in V1, if not already in the LGN. It is likely that stereo processing involves simple cells with non-zero phase characteristics (Ohzawa *et al.* 1997; Read & Cumming 2006).

V1 is composed of at least nine major layers, but the processing in those layers is not yet well understood. For nice overviews see Hubel (1995) and Schmolesky (2000).[1] Apart from simple, complex and end-stopped cells there also are bar and grating cells. These are specialised for extracting aperiodic bars and periodic gratings. In contrast to simple and complex cells, which can be seen as linear filters because they respond to all patterns, bar and grating cells are highly nonlinear: a bar cell does not respond to bright or dark bars in a periodic grating and a grating cell does not respond to isolated bars; see du Buf (2006) for a computational model of these cells and texture coding. There also are cells that respond to illusory contours, e.g., gaps in edges, for example caused by occluding objects like tree branches in front of other branches (von der Heydt *et al.* 1992; Heitger *et al.* 1998). Without doubt, there remain cells with other specific functions that will be discovered in the near future.

The tuning of cells to different frequencies (scales), orientations and disparities, together with the existence of e.g., bar cells, points at a multi-scale image representation: lines, edges, keypoints, gratings etc. It is even possible that disparity is attributed to extracted lines and edges, i.e., in principle it is possible to construct a 3D “wireframe” model of objects, like the solid models used in computer graphics, but this is still speculative. However, it is likely that there are at least three (interconnected) data streams within the ‘what’ and ‘where’ streams:

(1) The multi-scale line/edge representation serves object segregation, categorisation and recognition, with coarse-to-fine-scale processing, the latter also being applied to disparity in order to solve the correspondence problem. We may assume that this stream is responsible for line/edge-related brightness perception (see below).

(2) The multi-scale keypoint representation serves Focus-of-Attention (FoA), a process that directs our eyes—and mental attention—to points with a certain complexity: it does not make much sense to fixate points in homogeneous image regions where there are no structures to be analysed. In combination with motion and other cues, like colour contrast, this stream could be the basic cornerstone of the ‘where’ stream (Itti & Koch 2001; Rodrigues & du Buf 2006c).

(3) Colour and texture are surface properties of objects, normally in homogeneous regions but also with global modulations like shading due to light sources (shape-from-shading) and/or the shape of 3D objects (shape-from-texture). This shape information complements disparity information. Since lines and edges are 1D transitions (1D singularities; keypoints are 2D singularities) without colour, colour is supposed to be “sampled” and represented in the CO blobs (but see below!).

---

[1] cf. http://webvision.med.utah.edu/VisualCortex.html .

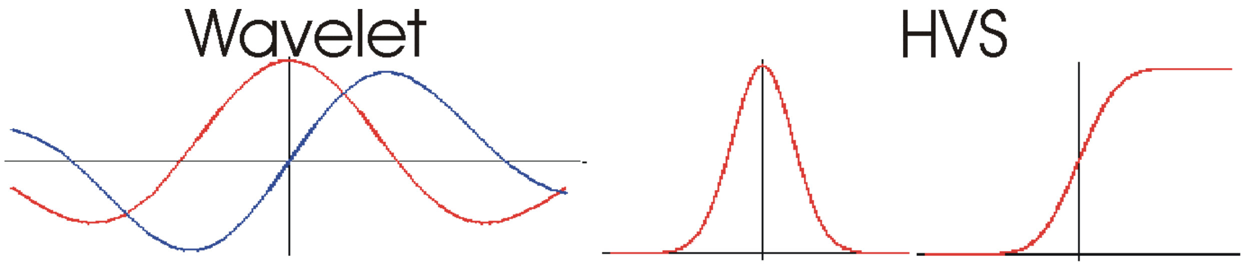


Figure 3: 1D cross sections of Gabor wavelets with sine and cosine components (left) and line and edge symbolic representations (right). A Gaussian window can truncate the error function at the far right.

This is an over-simplification of course, because FoA in textured regions can direct attention for scrutinising detail, i.e., a conscious action that may complement an unconscious process like automatic texture segregation, and global modulations (shape-from-X) can invoke different analyses. It is therefore important to stay focused on the main themes: basic processing serves (a) object structure, (b) surface structure, and (c) scene structure. Coming back to brightness processing, our model was conceived from three rather simple—not trivial—observations that are not so easy to explain to non-specialists:

(1) Simple cells are often modelled by complex Gabor (wavelet) functions, or quadrature filters with a real cosine and an imaginary sine component, both with a Gaussian envelope (see Fig. 3 (left), and du Buf (1993)). Such filters have a bandpass characteristic: the integral over the sine component is zero and the integral over the cosine component is very small or residual. Wavelets are also being used in image coding: the use of a complete set of bandpass filters tuned to all frequencies and orientations, plus one isotropic lowpass filter, which sum up to an allpass filter (a linear filter that passes all frequency components), allows to reconstruct the input image. Therefore, in principle the brain could use the same strategy: sum the activities of all simple cells plus one “lowpass channel,” for example from the special retinal ganglion cells with photoreceptive dendritic fields, if available in the CO blobs, into a retinotopic projection map in some neural layer. However, this leads to a paradox: it would be necessary to construct “yet another observer” of this map in our brain. Therefore, we assume that brightness is related to the multi-scale line/edge representation, which is necessary for object recognition.

(2) Basic line and edge detection involves simple cells in phase quadrature: positive and negative lines and edges (1D cross sections) can be detected and classified by combining detectors of zero-crossings and extrema (positive or negative) of the sine and cosine components, in combination with (positive) extrema of activities of complex cells. Our previous (van Deemter & du Buf 2000) and recent (Rodrigues & du Buf 2006a) models are based on simple and complex cells and are multi-scale, since many spatial patterns cannot be described using only one or few scales. However, there is one complication: at ramp edges, where a linear ramp meets a plateau, for example in trapezoidal bars or gratings

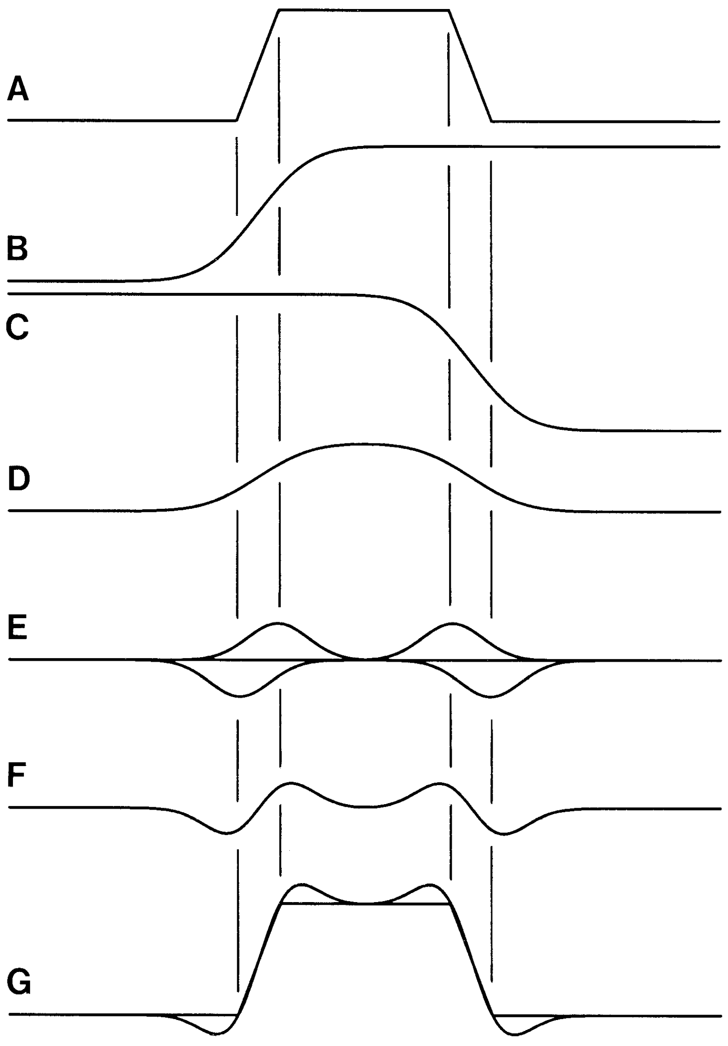


Figure 4: Mach bands at a trapezoidal luminance bar (A) can be explained by the multi-scale line/edge representation. At a very coarse scale a wide bar is detected (not shown here). At medium scales the two edges are represented by scaled error functions (B,C) which, when summed, also form a wide bar (D). At fine scales the four ramp edges are represented by positive and negative lines (E), which when summed (F) and combined with signal D create the typical overshoots (G).

(Fig. 4), the system will detect positive and negative lines. Responses of filters in quadrature do not allow distinguishing between lines and ramp edges, which explains Mach bands at ramp edges (du Buf 1994).

(3) The implicit, multi-scale line and edge representation must provide information for brightness construction by means of an interpretation. In other words, instead of a reconstruction the system builds a virtual impression on the basis of a learned interpretation of responding line and edge cells, perhaps much like a trained neural network. We "simply" assume that a responding line cell (at a certain position, tuned to a scale and orientation) is interpreted as having a Gaussian cross-profile there, with a certain amplitude (the response of the complex cell) and width (the scale of the underlying simple and complex

cells). The same way responding edge cells are interpreted, but with a bipolar (positive-negative) cross-profile and modelled by a Gaussian-windowed errorfunction (see also Fig. 10 in du Buf (1994)). Figure 4 illustrates the process in the case of a trapezoidal bar: the entire bar is represented by a very broad vertical line at coarse scales, by positive and negative edges at the ramps at medium scales, and positive and negative lines at the ramp edges at fine scales. If all is combined, the detected lines at fine scales cause Mach bands.

This model provides a completely new way for image (re)construction, not like coding based on wavelets or simple cells. An additional observation is that there is a lot of neural noise in the system and we do not know whether there exist simple and complex cells etc. at all retinotopic positions and tuned to all scales and all orientations (representation noise and completeness). Stained maps of hypercolumns and dendritic/axonal fields of most if not all cells look rather random (Hubel 1995). Nevertheless, the image that we perceive looks rather stable and complete. It is very simple to simulate what happens when we suppress information, both in the brightness model as described above and in wavelet coding, the latter being modelled by considering the summation of responses of simple cells. For example, we can suppress one entire scale channel, or 50% of all information by a random selection. Figure 5 shows what happens: the result is a very graceful degradation in the case of the brightness model, but a very disturbing rippling in the case of wavelet coding. This rippling in image coding requires sophisticated post-processing to reduce the effect, see for example Ye et al. (2004).

In the meantime the two questions at the end of Section 2.4 have been addressed: (a) The image that we perceive is a virtual construction by a symbolic line and edge interpretation, i.e., it is not a re-construction with no need for “yet another observer” in our brain who must analyse the reconstructed image for object recognition etc. In fact, object recognition and brightness perception have been combined into a single process: indeed, our simulations showed that object categorisation and recognition can be obtained by using different multi-scale image representations, i.e., either line/edge maps with event positions and types, or by the unimodal line and bimodal edge representations (Rodrigues & du Buf 2006a,b). (b) There is no precise region in our brain where the image that we perceive is created. Our model is limited to feature extractions in V1 and beyond, but this information must propagate to higher brain regions, eventually leading to consciousness, at the least being aware of our position in our actual environment. In other words, we may say that our perceived image, and therefore also at least part of our consciousness, are constructed by the entire brain, perhaps with an emphasis on the visual cortex. This is a holistic view, but it should be mentioned that the local-global discussion about consciousness might be a hornets’ nest (Koch 2004; Bauer 2004).

Figure 5: Image coding based on wavelets (left) and the brightness model (right). In both cases a limited number of scales has been used (top), which leads to severe rippling in the case of coding. If from all information only half is randomly selected, the coding result further deteriorates (bottom-left) but not the brightness result (bottom-right)

Above we wrote that colour is represented in the CO blobs in V1, possibly in the form of sampled values that represent homogeneous object regions. However, recently it was found that many colour cells in V1 are orientation tuned (Friedman *et al.* 2003). This probably means that such oriented edge (contour) cells also contribute to colour perception and not only to achromatic brightness as exploited in our brightness model. In addition, contour processing may play an important role in colour constancy, with different weights of near and far (local and global) contour components in the normalisation process, in addition to near and far colour samples; for a computational model see for example Rizzi *et al.* (2003). It should also be added that part of all neural connections may be more static and a result of evolution, i.e., brightness as an ecological interpretation of learned patterns in natural images (Yang & Purves 2004). All such complications, including long- and short-term adaptation effects and input-output amplitude nonlinearities, which have not even been mentioned until here, make us realise that we are far away from a unified framework.

The same can be said about object categorisation and recognition. Change blindness, the fact that we do not notice things at positions where we are not looking, points at an interpretational filling-in process. Even the filling in of the blind spots in the retinas, where the two optical nerves leave the eyes and there are no photoreceptors, is not noticed under normal viewing conditions. The latter effect could at least be explained by the fact that input from the other eye might be used there, but not change blindness. If we do not perceive a specific object, we do not perceive that object's brightness and colour. In such a case our brain may be guessing what the most obvious solution might be, probably on the basis of prior experience with similar images.

4 Painterly rendering

It is relatively straightforward to develop a painterly-rendering scheme on the basis of our brightness model, i.e., human vision, as is the case in similar approaches using algorithms from computer vision (Gooch *et al.* 2002; Kovács & Szirányi 2004; Shiraishi & Yamaguchi 2000). In our case, the scale of simple and complex cells is translated into the width of discrete brush strokes: single strokes in the case of detected lines and two parallel strokes in the case of detected edges, simulating coarse-to-fine painting using increasingly smaller brushes. Detected line and edge positions are stored in coordinate lists and these can be processed, for example smoothed, broken up into smaller lists, and/or linearised. For each coordinate list the stroke(s) is (are) rendered by means of triangle lists and texture mapping, for which colours are picked in the input image: one colour at the centre of line strokes and two colours at the centres of edge strokes. Texture mapping allows to simulate real brush strokes, composed of random selections of heads, bodies and tails of digitised strokes that were painted with a flat brush and, e.g., oil paint.

In homogeneous regions, where no lines and edges have been detected, we can prepare a background by applying strokes randomly or by influencing orientations for diagonal (or rotated) criss-crossing. In fact, we always start with painting a complete background, like most painters do, because our interface allows to select line/edge-related foreground strokes with certain brush sizes. The use of all scales and therefore brush sizes will result in a very realistic painting; when some scales are skipped the result will be more abstract. In addition, when introducing an orientation bias, i.e., for example rotating brush strokes towards horizontal, vertical and diagonal orientations, the result will become more cubistic with increasing bias.

The user interface which is being developed has very few menu lists and a structure that resembles the procedure that a painter uses: first select a surface structure (canvas or paper) and background colour, then apply a background with random or biased strokes, which can be incomplete because the user can stop the painting process at any time, for example to adjust parameters. To this end the user can set the speed of the painting process, can stop, resume or re-start the entire process or only the back-or foreground process. The interface allows to apply palette effects, for example to apply a model of colour constancy—a sort of normalisation of the dynamic ranges of the R, G and B channels—

which normally makes a painting more vivid, and/or to apply a red-orange or blue-green shift for introducing a warm or cold emotion. The interface also allows to apply a model of Focus-of-Attention based on end-stopped cells, in order to apply brush strokes only in and around regions with some complexity. Figures 6 and 7 show a few examples. For further details we refer to du Buf *et al.* (2006) and Nunes *et al.* (2006). Future research goals are to study the influence of colour shifts, not only for colour emotions, and the level of image abstraction, in simulated paintings. Such aspects are closely related to painting styles and studied in a research area called empirical aesthetics. Image and painting composition is much harder to address in terms of the visual cortex, although simple manipulations of existing paintings have been applied in some studies, see Nodine *et al.* (2003) and Locher (2003).

5 Discussion

In the Introduction we wrote that we cannot take a look into van Gogh's head and we do not know the exact landscapes that he saw. Well, after reading the subsequent sections the reader should be able to assume that we are on the way to simulate a standard observer in conjunction with a standard painter. In other words, we start being able to explore basic processes in the visual system and to combine these into an increasingly complete architecture, thereby implicitly looking into a "generic head" with the possibility to simulate specific painters in the future.

The visual system is able to construct on the basis of a brief glance a complete image/scene representation in our brain: from local syntax to objects to gist and layout of objects, including semantic interpretations and even emotions. More advanced models will therefore lead to a complete morphology, as if someone is asked to write a complete description of an image, from global aspects to local detail. Unfortunately, the development of a complete artificial visual system—or computational model—is a very long-term goal. However, the image interpretation, description, annotation etc. are expected to foster novel solutions for image and video synthesis, coding and art work for illustration purposes. The development will depend on results of ongoing and future research projects, both in visual perception and in NPR. Since even relatively simple models of the visual system require tremendous amounts of storage capacity and associated CPU times for the number crunching, new generations of more powerful computers are required. As for now, we do not know whether parallel processing in a distributed Grid environment will be beneficial because of necessary communication times, but the tremendous storage capacity that is required is no surprise: the entire brain counts 1012 (one million million) cells with 1014 to 1015 interconnections, and a significant part is devoted to vision. Today, in 2007, it is already possible to achieve 1 TFLOPS (one tera or one million million of floating point operations per second) on a normal PC using graphics boards with GPUs that are optimised for vectorised MADD (multiply-add) operations. This is not a supercomputer, but on compara-

Figure 6: Rendering: the input image (top-left) is first used to paint a background with a big brush (shown on the third row at left), on which foreground strokes can be painted using increasingly smaller brushes. Not all scales need to be painted

ble systems it will soon be able to simulate the dynamics of 1012 cells at a speed which will come close to realtime, provided that enough of fast memory is available. Storage capacity being the bottleneck, future hard disks with a capacity of more than 1 TBYTE will not provide a solution because of slow access times.

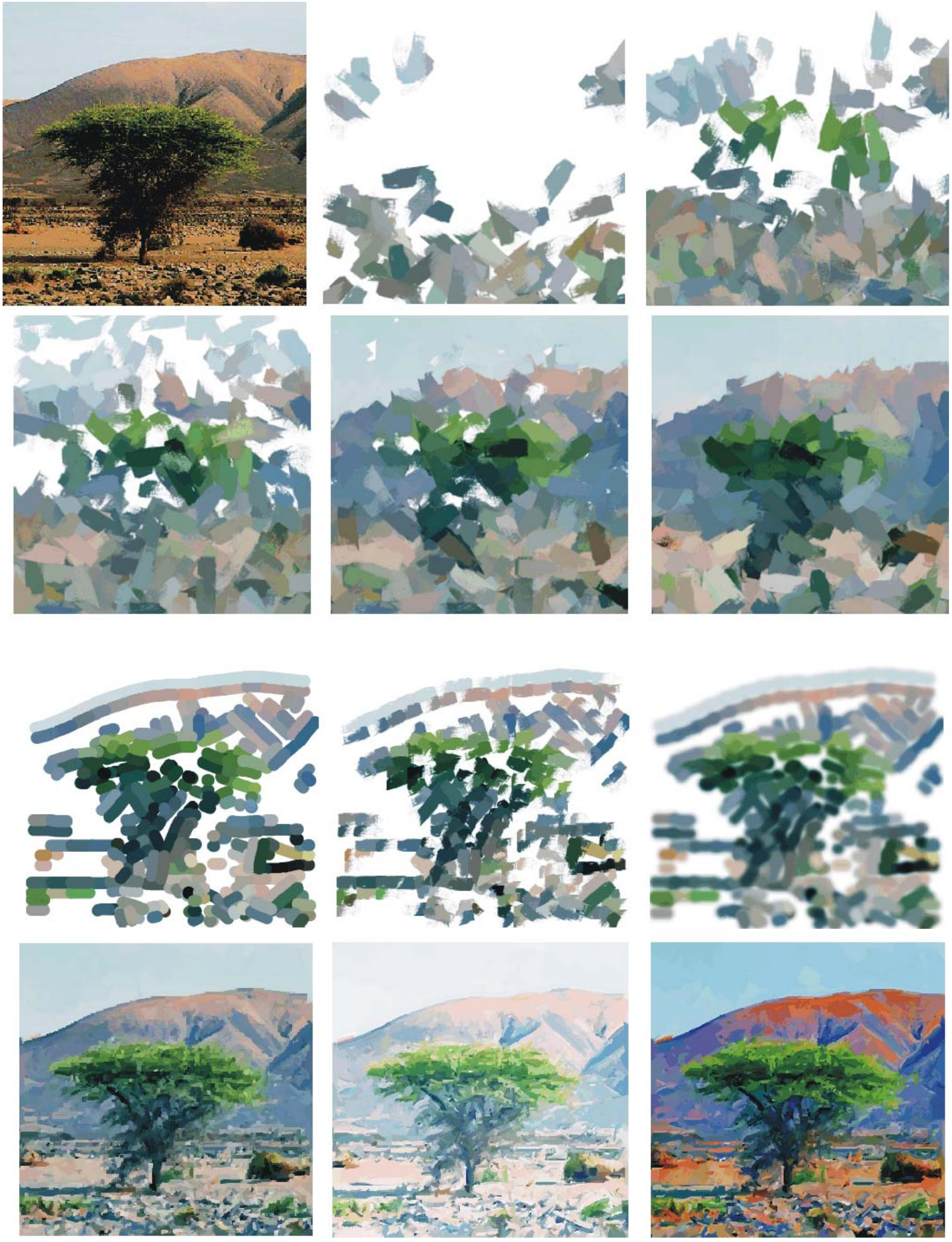


Figure 7: Top two rows: input image and the background process with random strokes of a flat brush. Third row: foreground strokes with a round brush, a flat one and spray. Bottom row: changing brightness and saturation

Although not discussed explicitly, it should be clear that our models provide a morphological image description in terms of multi-scale features on the basis of cortical cells: lines and edges for brush strokes and keypoints for Focus-of-Attention. Future extensions will cope with abstraction and composition, even with meaning or gist. All these features can be extracted by data-driven or bottom-up processes. So why could we write in the abstract that bottom-up and top-down processing can be combined?

The answer is rather straightforward: one might say that our visual system has two modes of operation. When looking at an image for a split second, long before we consciously know what objects there are, our brain already knows what the image is about. This is the fast gist and layout vision, probably implemented by feed-forward neural networks that exploit texture, colour, disparity and motion. Such features also allow for separating (segregating) entire objects, for example a tree with differently coloured and textured trunk and crown in front of a background, where trunk and crown should belong to the same object. Hence, in addition to global gist there may exist local gist, which hints at specific objects and their spatial relations (layout). This first and rapid mode of operation can be thought to "bootstrap" the second mode: select subsets of normalised templates in memory in order to scrutinise objects in the input image. The latter objects are not normalised, which implies that multi-scale line/edge and keypoint representations of input objects and normalised templates in memory must be compared. This comparison must be done sequentially, object after object, and the two feature maps must be projected such that they converge. This is the dynamic linking between neural layers at low and high levels as explored by Deco & Rolls (2004), and the fact that a big part of the visual cortex is involved in the dynamic linking limits the "bandwidth" of the system (Rensink 2000).

So, why is our visual system so fast and efficient? Because bottom-up and top-down processing are done in parallel. We do not think in terms of object edges or textures, we think in terms of gist, and gist limits the enormous amount of possible object templates in memory that must be checked. This explains why we have difficulties in recognising objects that are completely out of their normal context. In conclusion, the good news is that bottom-up and top-down image morphologies can or even must be combined. The bad news is that we are much more advanced in bottom-up processing, i.e. top-down processing is an almost completely new research area. However, in one or a few decades from now, when a lot of research effort has been put into top-down processing, this bad news will turn into good news for image morphology!

References

Allman, J., Zucker, S.: Cytochrome oxidase and functional coding in primate striate cortex: a hypothesis. Cold Spring Harbor Symp. on Quant. Biol. 55, 1990, 979–982.

Bar, M.: Visual objects in context. *Nature Reviews: Neuroscience* 5, 2004, 619–629.

Bar, M. & Kassam, K. & Ghuman, A. & Boshyan, J. & Schmid, A. & Dale, A. & Hämäläinen, M. & Marinkovic, K. & Schacter, D. & Rosen, B. & Halgren, E.: Top-down facilitation of visual recognition. *Proc. National Academy of Science* 103 (2), 2006, 449–454.

Bauer, R.: In search of a neuronal signature of consciousness – facts, hypotheses and proposals: neuroscience and its philosophy. *Synthese* 141 (2), 2004, 233–245.

Berson, D. Strange vision: ganglion cells as circadian photoreceptors. *TRENDS in Neurosciences* 26 (6), 2003, 314–320.

Deco, G. & Rolls, E.: A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Res.* 44, 2004, 621–642.

du Buf, J.: Responses of simple cells: events, interferences, and ambiguities. *Biol. Cybern*. 68, 1993, 321–333.

du Buf, J.: Ramp edges, Mach bands, and the functional significance of the simple cell assembly. *Biol. Cybern*. 70, 1994, 449–461.

du Buf, J.: Modeling brightness perception. Chapter 2 in: van den Branden Lambrecht, C.J. (ed.): *Vision models and applications to image and video processing*. Kluwer Academic, 2001, 21–36.

du Buf, J.: Improved grating and bar cell models in cortical area V1 and texture coding. *Image and Vision Computing*, doi:10.1016/j.imavis.2006.06.005, 2006 (Vol. 25, 873-882, 2007).

du Buf, J. & Fischer, S.: Modeling brightness perception and syntactical image coding. *Optical Eng.* 34 (7), 1995, 1900–1911.

du Buf, J. & Rodrigues, J. & Nunes, S. & Almeida, D. & Brito, V. & Carvalho, J.: Painterly rendering using human vision. *VIRTUAL - Advances in Computer Graphics in Portugal*, p. 12 (http://virtual.inesc.pt/), 2006.

Friedman, H. & Zhou, H. & von der Heydt, R.: The coding of uniform colour figures in monkey visual cortex. *The Journal of Physiology* 548, 2003, 593–613.

Gooch, B. & Coombe, G. & Shirley, P.: Artistic vision: painterly rendering using computer vision techniques. Proc. ACM/SIGGRAPH-Eurographics NPAR, Annecy (France), 2002, 83–91.

Heitger, F. & von der Heydt, R. & Peterhans, E. & Rosenthaler, L. & Kubler, O.: Simulation of neural contour mechanisms: representing anomalous contours. *Image and Vision Computing* 16 (6), 1998, 407–421.

Hubel, D.: *Eye, Brain and Vision*. Scientific American Library, 1995.

Itti, L. & Koch, C.: Computational modeling of visual attention. *Nature Reviews Neuroscience* 2 (3), 2001, 194–203.

Kastner, S. & Schneider, K. & Wunderlich, K.: Beyond a relay nucleus: neuroimaging views on the human LGN. *Progress in Brain Res*. 155, 2006, 125–143.

Koch, C.: *The Quest for Consciousness. A Neurobiological Approach*. Roberts and Company Publishers, 2004.

Kovács, L. & Szirányi, T.: Painterly rendering controlled by multiscale image features. Proc. 20th Spring Conf. Comp. Graphics, Budmerice (Slovakia), 2004, 177–184.

Locher, P. E.: Experimental aesthetics: the state of the art. *Bulletin of Psychology and the Arts* 4 (2), 2003.

Nodine, C. & Locher, P. & Krupinski, E.: The role of formal art training on perception and aesthetic judgment of art compositions. *Leonardo* 26 (3), 2003, 219–227.

Nunes, S. & Almeida, D. & Brito, V. & Carvalho, J. & Rodrigues, J. & du Buf, J.: Perception-based painterly rendering: functionality and interface design. Proc. Ibero-American Symp. Comp. Graphics, Santiago de Compostela (Spain), 2006, 53–60.

Ohzawa, I. & DeAngelis, G. & Freeman, R.: Encoding of binocular disparity by complex cells in the cat's visual cortex. *J. Neurophysiol*. 18 (77), 1997, 2879–2909.

Pessoa, L.: Mach bands: how many models are possible? Recent experimental findings and modeling attemps. *Vision Res*. 36, 1996, 3205–3227.

Read, J., Cumming, B.: Does depth perception require vertical-disparity detectors? *Journal of Vision* 6 (12), 2006, 1323–1355.

Rensink, R.: The dynamic representation of scenes. *Visual Cogn.* 7 (1-3), 2000, 17–42.

Rizzi, A. & Gatta, C., Marini, D.: A new algorithm for unsupervised global and local color correction. *Pattern Recogn. Lett.* 24 (11), 2003, 1663–1677.

Rodrigues, J. & du Buf, J.: Cortical object segregation and categorization by multi-scale line and edge coding. Proc. Int. Conf. Comp. Vision Theory and Applications 2, Setúbal (Portugal), 2006a, 5–12.

Rodrigues, J. & du Buf, J.: Face recognition by cortical multi-scale line and edge representations. Proc. Int. Conf. Image Anal. Recogn., Springer LNCS Vol. 3211, 2006b, 329–340.

Rodrigues, J. & du Buf, J.: Multi-scale keypoints in V1 and beyond: object segregation, scale selection, saliency maps and face detection. BioSystems 86, 2006c, 75–90:doi:10.1016/j.bio systems. 2006.02.019.

Schmolesky, M.: The primary visual cortex. http://webvision.med.utah.edu/VisualCortex.html, 2000.

Shiraishi, M. & Yamaguchi, Y.: An algorithm for automatic painterly rendering based on local source image approximation. Proc. ACM/SIGGRAPH-Eurographics NPAR, Annecy (France), 2000, 53–58.

van Deemter, J. & du Buf, J.: Simultaneous detection of lines and edges using compound Gabor filters. *Int. J. Patt. Recogn. Artif. Intell.* 14 (6), 2000, 757–777.

von der Heydt, R. & Peterhans, E. & Dursteler, M.: Periodic-pattern-selective cells in monkey visual cortex. *J. Neurosci*. 12 (4), 1992, 1416–34.

Wässle, H.: Parallel processing in the mammalian retina. *Nature Rev. Neuroscience* 10, 2004, 747–757.

Yang, Z., Purves, D.: The statistical structure of natural light patterns determines perceived light intensity. Proc. Nat. Acad. Sci. USA 101 (23), 2004, 8745–8750.

Ye, S., Sun, Q. & Chang, E.: Edge directed filter based error concealment for wavelet-based images. Proc. IEEE Int. Conf. on Image Processing 2, 2004, 809–812.

Zucker, S. & Hummel, A.: Receptive fields and the representation of visual information. *Human Neurobiol.* 5 (2), 1986, 121–128.

[Table of Content]

The SVP Group[‡]

Automatic Generation of Movie Trailers using Ontologies

Abstract

With the advances in digital audio and video analysis, automatic movie summarization has become an important field of research. Much of the work has been put into movie abstracting for large media databases. Looking at the topic from a different side, the movie industry has long since perfected the art of summarization in their advertising trailers to attract an audience. In this paper we introduce the approach of automatically generating entertaining Hollywood-like trailers based on a trailer grammar, enhanced by an ontology. The extraction of features from movies using state-of-the-art image and audio processing techniques builds the foundation for the selection of meaningful and usable material, which is re-assembled according to the defined rules. User testing of our automatically produced trailers shows that they are well accepted and in many ways comparable to professionally composed trailers.

1 Introduction

Although originally intended to advertise a certain movie, the short preview of a movie, i.e., trailer or teaser, has become an attractive movie genre in itself [Kernan 2004, Arijon 1991], especially since many trailers are available on the Internet. With the development of current digital technology the question arises if and to what extent it is feasible to automate the process of trailer production based solely on a high-level analysis of the original movie. Such a system could provide improvements in different movie-related fields. For example, it could suggest innovative ways of video browsing in digital movie databases in a way that those trailers could serve as a compact overview of a certain movie or to gain more control especially in movie-on-demand system through movie indexing, retrieval, and browsing

---

[‡] *The SVP-Group is in alphabetical order: Christoph Brachmann, Hashim I. Chunpir, Silke Gennies, Benjamin Haller, Thorsten Hermes, Otthein Herzog, Arne Jacobs, Philipp Kehl, Astrid P. Mochtarram, Daniel Möhlmann, Christian Schrumpf, Christopher Schultz, Björn Stolper, and Benjamin Walther-Franks.*

(see [Chen *et al.* 2004]). Furthermore, it could help with developing and testing of experiments to formalize existing movie editing methods (film theory), and simplify or even extend the work of editors/authors (see [Snoek & Worring 2005]).

In this paper, an automatic trailer generation system is introduced which covers three research challenges. First, a formalism is proposed that describes the basic components of a trailer. Second, video content analysis methods are presented that provide these single components. Finally, a methodology is shown that selects and composes the components according to the given formalism.

This paper is organized as follows. In section 2 a brief overview about previous work related to automatic trailer generation is given. Section 3 describes the formalism that we developed and use as a basis for our system. This approach is based on an ontology, and according to [Chen *et al.* 2004] one could say it is like a grammar for an automatic trailer generation, but we consider our approach as rule-based. Section 4 illustrates our system framework that – based on the given set of rules – is capable of analyzing a movie first, and then is able to generate trailers of this analyzed and annotated movie. In section 5 we present the application of our system to generate trailers for current Hollywood action movies along with an evaluation of the corresponding output in section 6. Finally, section 7 draws a conclusion and addresses possible aspects of future work.

2 Related Work

In this section we provide a very brief overview of touching approaches since the specific way of generating movie trailers has only little related work so far. One can say that area of automatic trailer generation is a rather untouched field of research. However, the more general task of summarizing video content is a wide field of research.

The works of [Chen *et al.* 2004] and [Lienhart *et al.* 1997] come closest to our goal. Both mention the possibility of generating a movie trailer explicitly. And furthermore, both point out to do the composition of footage according to rules derived from film theory and present ways to retrieve crucial information for trailer generation. But they do not focus on how to compose trailers. Only [Chen *et al.* 2004] uses the definition of *tempo* in order to generate action trailers. Although the film theory is valuable in the analysis of the footage deriving high-level features from low level features, it is not completely applicable for the generation of a movie trailer.

Since a movie trailer is a kind of abstract of a movie other works within the field of video abstracting or summarization rather focus on the task of pure summarizing in order to provide means to handle the increasing amount of video data. One could find three basic approaches. The first one is video skimming as, e.g., in [Christel *et al.* 1999] or [Smith & Kanade 1998] where video material is analyzed and condensed to important scenes. Typically the linearity of the input video is preserved. The second basic approach is summarizing contents in a pictorial way [Uchihashi *et al.* 1999, Yeung & Yeo 1997]. In [Uchihashi *et al.* 1999] in a first step salient single frames of video sequences are captured. In a second step these frames are sized according to their importance, and finally arranged in a third step in a linear comic-like, story-telling way. The third video browsing approach is closely related to the pictorial summarization but focuses on a hierarchical, not necessarily linear way of presenting the video content [Ponceleon & Dieberger 2001, Zhang *et al.* 1993].

The degree of automation varies. Completely automatic approaches are [Lienhart *et al.* 1997, Smith & Kanade 1998, Uchihashi *et al.* 1999]. Typical for these approaches of automatic summaries is the high dependency on low level analysis of image and audio. A so-called semi-automatic approach can be found in [Zhu *et al.* 2003]. The semi-automatic summary tools provide some manual annotation framework enabling high-level analysis to conclude what is happening in a scene. Another interesting work is [Ma *et al.* 2002], focusing on the question of how a video is perceived by a user.

Finally, while some works – [Lienhart *et al.* 1997, Smith & Kanade 1998, Zhu *et al.* 2003] – can be applied to a wide variety of footage, others focus on a specific type of video data, e.g., sports [Babaguchi *et al.* 2005].

An extensive overview of “State-of-the-Art” video indexing from the author’s point of view can be found in [Snoek & Worring 2005].

3 Formalism

In the scope of re-assembling movie footage in a short video that can be labeled as a *trailer*, first the meaning of this label has to be understood. According to [Arijon 1991] films are created based on an underlying *Film Grammar* to successfully communicate with the audience. In [Kernan 2004] it is stated that a trailer is also a movie genre of its own right. Therefore we assume that trailers - being a special kind of film – can be described by syntactic elements and semantic rules. One will say that this constitutes a *trailer grammar* but we will stay to a rule-based system. To implement an automatic trailer generator these rules have to be understood and modeled in a way that a computer can execute generative algorithms according to them. The problem therefore demands understanding, extracting, and formalizing two items: the trailer’s syntactic elements (section 3.2), and its semantic rules (3.3). Before looking at these, we give a definition of the term ‘trailer’ within the context of automatic generation.

3.1 The Definition of a Trailer in Respect to Automatic Generation

[Kernan 2004] points out that the name *trailer* refers to the fact that these short movies were originally shown at the end of a film program in movie theaters. But nowadays trailers were normally shown before the main movie. During the 20th century, trailers evolved from a mere advertisement to a movie genre with its own unique conventions, based on the demand to combine an artistic form with the highly commercial need of drawing the biggest possible audience into the theaters by presenting every movie in the most attractive light.

While movies and trailers exist in many different forms according to different cultural environments, our automatic approach is based on the Western culture's most dominant trailer and movie industry: Hollywood blockbuster cinema. Although trailers from this domain have developed a general formula that pays as little attention to genre or specific target groups as possible – to attract literally everybody and lead to an "undifferentiation of audiences" – they still have to reflect the movie in question to a certain extent [Kernan 2004]. Furthermore, our aim is to produce short videos that resemble rather conventional *Theatrical Trailers* by having a length of more than one minute and featuring footage from the original movie. These are opposed to so-called *Teaser Trailers*, which are typically produced before primary shooting is finished and consist mostly of texts, voice-overs, and graphic elements, and which have a maximum running length of one minute. In the following the term *trailer* therefore will be used referring to a theatrical trailer for a contemporary Hollywood movie.

3.2 The Syntactic Elements of a Trailer

The basic elements of any edited movie are usually shots and transitions. We assume that within these elements certain types of shots and transitions can be identified by a shot-by-shot analysis of original movie trailers. In order to determine these types, an appropriate set of descriptions, i.e., an appropriate vocabulary, has to be defined. Since our goal is to implement a completely automatic system for trailer generation, we consider the restrictions imposed by the technical feasibility when setting up such a vocabulary. This inevitably causes a quality loss but cannot be avoided in our case. Furthermore, there is a trade-off concerning the level of detail when defining the appropriate descriptions for shots. If the detail is too high, the shot descriptions are only suitable for a very specific situation in one trailer. On the other hand, if the level of detail is too low, they are too general and have very little meaning. That is why the resulting types of shots have to:

a) be able to cover all shots of a trailer,
b) be clearly distinguishable from each other (no redundancy),

c) have a well-defined meaning,
d) apply to as many existing trailers as possible, and
e) be defined based on the information which will be extracted from the movie by our automatic analysis tools (technical feasibility).

Besides this, describing the transitions is easier since they follow the known conventions of the film grammar. Well-known transitions are for example *hard-cuts*, *fade-ins*, and *fade-outs*.

In order to distinguish between the original movie and trailer shots, and the shots we produce for our trailers we refer to the latter ones as *clips*, and in order to fulfill the requirements listed above we define the types of the *clips* by the following *properties*:

- a **category** (reflecting the shot's formal features),
- the playback **speed** (to model effects like slow-motion or acceleration),
- the **volume** of the original footage sound (so that clips can be muted or amplified),
- and **location**, corresponding to the footage location in the source movie.

3.3 The Semantic Elements of a Trailer

Once *clips* and *transitions* are identified and described as syntactic elements of a trailer, semantic rules are needed to assemble these elements in a trailer-like way. We propose to represent these rules as a hierarchy of super- and sub-patterns as shown in Figure 1. Each super-pattern consists of a number of sub-patterns either in a certain order or as a random choice.

The highest level of patterns is the *Trailer Pattern*. Since there is not only one universal pattern that can describe all trailers at once, this pattern can be used to distinguish between different types of trailers. For example, one Trailer Pattern could stand for action movie trailers, and another Trailer Pattern could stand for a romantic movie trailer.

In our model every trailer can be subdivided into a number of different narrative blocks, which we call *Phases*. These *Phase Patterns* could one of the five following phases we identified in contemporary trailers: *Intro*, *Story*, *Break*, *Action*, and *Outro* (see also section 5.1).

The Phase Patterns again are composed of *Sequence Patterns*, which in turn consist of a number of *Clip/Transition Pairs*. These Pairs are the lowest level of the hierarchy. Therefore, a trailer is described by a linear list of clips joined by transitions. The intention behind the modeling of a trailer in such a way is to represent the trailer grammar as precise as possible, while preserving the highest possible amount of flexibility.

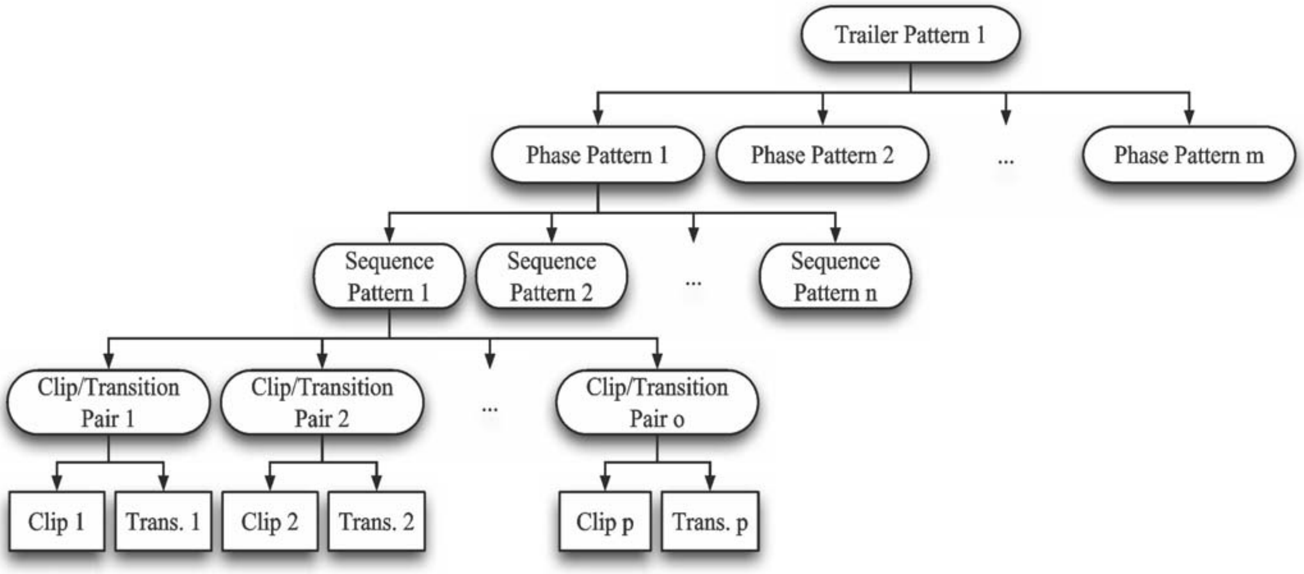


Figure 1: A branch of the hierarchical view of a generic trailer structure

4 System Framework

Our system framework comprises two major components. The first one is a collection of various low- and high-level image and audio processing modules, which provides information about a given movie by extracting a set of features. Each analysis module is described in the following sections 4.1.2 to 4.1.13. The second component provides an implementation of the proposed trailer rule base, which is described in detail in sections 4.2.1 to 4.2.6. This component is able to categorize the annotated information of the first component and to use that data to automatically assemble a full trailer.

4.1 Extracting and Annotating Movie Features

In order to extract features of a given movie we use on the one side methods of image and audio analysis on different levels of abstraction, and on the other side we derive data from Internet resources. By combining the output of several modules with each other we extend the complexity and reliability of the annotated data. Figure 2 gives an overview of the interdependencies among the single modules.

As a ground truth we manually annotated the movie *The Transporter (2002)* and we adapted the performance scale of precision $\underline{\boldsymbol{P}}$ and recall $\underline{\boldsymbol{R}}$ to frame ranges as following

$$\underline{\boldsymbol{P}} = \frac{\text{number of relevant frame ranges retrieved}}{\text{number of frame ranges retrieved}}$$
$$\underline{\boldsymbol{R}} = \frac{\text{number of relevant frame ranges retrieved}}{\text{number of relevant frame ranges}} \qquad (1)$$

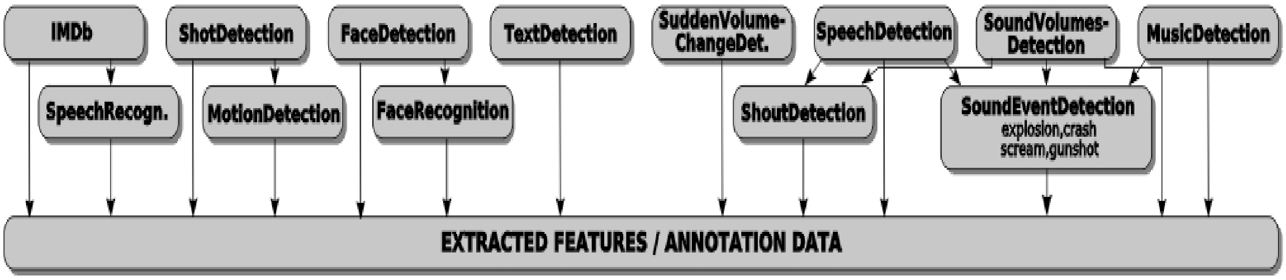


Figure 2: Overview of the modules for extracting features and their dependencies on each other

Therefore, we have a basis for evaluating the output of every module on basis of the approach of information retrieval. In the following section each single module is described along with the corresponding movie feature(s) it is providing.

4.1.1 Internet Resources

The Internet resources of the *Internet Movie Database*[1] (IMDb) are used to augment and enhance the generation of the trailer with automatically extracted data such as movie title, director, actors, genre, awards won and production company, which are used to generate credits for the trailer. In addition, famous quotes of the selected movie are extracted and used to perform a keyword-spotting in the speech recognition module.

This module is realized as a Python script using the IMDbPY[2] package to retrieve and manage the desired data of the IMDb.

4.1.2 Shot Detection

In order to detect shot boundaries we use an existing tool that was developed by other members of our research group [Miene *et al.* 2001]. However, our system just incorporates the shot boundary detection by the Gray Histogram X^2 Feature extraction so only hard cuts are detected. We set the adaptive threshold $Th_{percentage}$ to 7 and the minimum frame number to 7, which results in precision and recall values of 0.93 each. In addition, we calculate average color values for each detected shot. In future, the precision and recall values should be improved by extracting other features and detecting other transition types, e.g., dissolves.

4.1.3 Motion-based Segmentation

We also segment a movie into frame ranges with homogeneous motion intensities. First, motion intensities are calculated for each pair of adjacent frames by the pyramidal imple-

---

[1] Cf. http://www.imdb.com

[2] Cf. http://www.imdbpy.sourceforge.net

mentation of the Lucas Kanade feature tracker provided by the OpenCV library[3]. Due to its pyramidal approach, it deals with small and large motions in a balanced way. After the calculation, we add up all feature motions except for the ones below a certain threshold $T_{minFeatMot}$ in order to disregard hardly noticeable motion. Furthermore, the calculated sum is reduced if the image brightness is low, as black frames tend to cause high motion intensity because of encoding artefacts. We also reduce this sum for a frame pair corresponding to a hard cut, as such a transition naturally causes high motion intensity.

Now, we use a grid of several motion classes in order to classify each frame pair motion according to its motion intensity per feature. Adjacent frames with the same class are then combined to frame ranges. In order to avoid too short motion frame ranges we define a minimum length. If a frame range has a length below this threshold T_{minLen} then it is combined with the neighboring frame range. The results of the motion-based segmentation run on nine test movies were very satisfying, particularly concerning frame ranges with very low and very high motion intensity, respectively. Future work should extend the movement information by differentiating between camera (zoom, pan, etc.) and object motion.

4.1.4 Face Detection

In order to find actor appearances within a movie, we use the basic face detection algorithm provided by the OpenCV library, which uses the Haarcascade classifier [Lienhart & Maydt 2002] for detection with a minimum of three neighbors used for grouping. After the detection process we cluster single detected faces to sequences so that we can define a frame range for an appearance of an actor. Variables considered during this clustering process are face deviation in size and position within a frame along with a threshold for closing gaps between successive frames caused by occlusions or head movement. For computing a mean face image and for the later definition of a subspace during the face recognition we use a publicly available face database[4]. The faces out of this database are all frontal faces in different lighting conditions without any rotation. We compute the distance of every face in a sequence to the mean image, i.e., to the mean face. The face that performs best is chosen to be the representative for the sequence. We assume that a small distance indicates a frontal face barely rotated. This distance along with face size in comparison to the frame width of the movie gives hints on close-up shots and not being a false positive.

The face detection achieves a precision of 0.8 and the distance to the mean image prevents many of the false positives of being used. However, future work should certainly involve a human skin detection to provide even more robust results.

---

[3] Cf. http://www.intel.com/technology/computing/opencv/

[4] Cf. http://www.equinoxsensors.com/products/HID.html

4.1.5 Face Recognition

For face recognition we decided on using Principal Component Analysis (PCA) as used in [Yambor 2000]. For the normalization process we implement a search for corners with big eigenvalues within an image to identify eye and mouth candidates. A problem arises, when only parts of a face are exposed to sunlight because they tend to produce stronger corners than parts left in shade. In order to compensate this, we impose a minimum distance between two points. We now search for strong corners as candidates with different distances imposed, compute a transformation matrix for every possible combination of candidates and apply it onto the face image. Our normalization outcome is a face image of 25x25 pixels with a mask applied on it occluding the background. We use 90 eigenfaces for the projection and do a k-means clustering on the results. We assume that the biggest cluster will be the cluster containing the first main actor, so that we achieve a precision of 0.59 and a recall of 0.16. Future work will be to implement a clustering that can deal with an unknown number of classes, and to produce a better homogeneity. Furthermore, it is necessary to improve the results of the face recognition by utilizing other techniques such as Elastic Bunch Graph Matching (EBGM; see [Wiskott *et al.*]1997).

4.1.6 Text Detection

The text detection is done by an existing tool of our research group first realized in [Wilkens 2003], which is specifically designed to find overlaid text in video. We use a 3x3 edge filter subtracting the lowest value from the center value for preprocessing. We then consider any group of more than three characters as text and choose rather strict settings in terms of deviation in tracking, horizontal spacing, vertical and horizontal scaling to keep the false positive rate as low as possible. This is important as shots containing text mostly end up in the black list during the generation process and we do not want to loose valuable shots. In case of our reference movie, the tool achieves a precision of 0.92 and a recall of 0.78.

4.1.7 Sound Volume-based Segmentation

Quiet portions of the movie will probably not contain action sequences but rather dialogs or scenery shots. Low volume can therefore be a very reliable indicator for falsely detected explosions, gunfire or other action-related elements. On the other hand, high volume can be a clue for action scenes, loud music or other noisy settings. One problem for the measurement of the audio volume is its quick fluctuation. It can vary significantly from one movie frame to the next. It is necessary to smooth the intensities over a range of many frames to get more stable and meaningful values. While smoothing the audio intensities and grouping them into frame regions it is desirable to assign these regions to portions of audio that maintain a relatively constant level. The borders to the neighboring regions should be placed wherever there is a significant change in audio volume. We propose to set the starting points for the regions to the points of minimal change in the smoothing function. The borders between the regions are then adjusted so that the error in respect to the region's average is minimized.

4.1.8 Sudden Volume Change Detection

A sudden increase in the loudness of movie audio is a clue for a deliberately integrated surprise element. We define such an increase as an extended period of quiet audio, e.g., one second, followed directly by a noisy part, where a high level of audio is sustained for another second. This definition makes sure that short bursts of loud audio will not be counted. For various movies, these sudden volume changes are often explosions, crashes or surprise attacks. However, not all volume increases are necessarily due to spectacular effect scenes. Sometimes the contrast of volume is used to emphasize the harsh cut from a quiet scene to a loud setting, such as to a disco or a factory hall.

4.1.9 Speech Detection

We perform a segmentation of the movie into speech and non-speech. A segmentation using the Bayesian Information Criterion (BIC) and zero-crossings as described in [Biatov & Köhler 2003] is only applicable on radio or TV broadcast news, consisting of clear speech, silence and music without overlay. For a good and reliable segmentation of Hollywood movies these methods give no valuable results. In our approach, a speech recognition is performed on the movie using the CMU Sphinx 3.5[5] speech recognition system in combination with the pre-trained open source HUB4 acoustic models and the small AN4 3-gram language model based on 130 words and numbers. Finding speech in the movie, even when disturbing music and background noise is present, works very well because only the found frame ranges are used and the content of the speech is not important.

After running the speech recognition, the extracted frame ranges represent single words recognized by the speech recognition system. In order to reduce false positives, all frame ranges F $(f_s < F < f_e)$ with $F > 18$ frames are removed. By combining frame ranges that have a distance between each other of $f_{2s} - f_{1e} < 50$ frames a complete dialogue structure can be formed. The evaluation of the accuracy of the system results in a precision of 0.79 and recall of 0.77 at a real time computation speed. Most of the false positives are singing artists in the background music.

---

[5] Cf. http://cmusphinx.sourceforge.net

4.1.10 Speech Recognition

The speech recognition module performs a keyword spotting to find the frame ranges that comprise the given famous quotes extracted from the IMDb by the Internet resources module. A typical speech recognition system uses phoneme-based acoustic models in combination with a word or syllable-based language model [Schrumpf *et al.* 2005]. In research, the most often used data is clearly spoken broadcast news without background noise or music. In contrast, Hollywood movies contain lots of overlays of speech, music, and special sound effects. In addition, some other factors like slang, blurring of word boundaries, strong variations in articulation, and speaker- or character-dependent characteristics make it even more difficult to achieve good results in this scenario.

Our module uses the CMU Sphinx 3.5 speech recognition system in combination with the pre-trained open source HUB4 acoustic models and a language model built out of the extracted quotes. By means of the CMU-Cambridge Statistical Language Modeling toolkit[6] the language model is built and the text-to-phone software addttp4[7] is used to build the word-phoneme dictionary. The difference to other language models is the fact that our language model uses each quote as one entity in the model and we only build uni-gram models, because there are no dependencies between single quotes.

After performing the speech recognition, the frame range with the highest probability is selected for every quote. This results in a precision and a recall of 0.67 each at a computation speed of 5 times real time. The result of the recognition highly depends on the quality of the IMDb quotes, which sometimes are not verbatim and thus cannot be found by the system. The next step to enhance the results would be to train acoustic models on manually annotated Hollywood movies. This would incorporate background noise and music into the acoustic models and make it more fitting for this domain.

4.1.11 Shout Detection

We also try to locate frame ranges in the movie where people shout by combining the output of the speaker detection and the sound volume-based segmentation. The program searches for frame ranges where the normalized sound volume v with $0 \leq v \leq 1$ exceeds a threshold $v_t \geq 0.5$. Only frame ranges of the speech detection that are at the same range as the thresholded sound volume are extracted. The recognition works with a precision of 0.5 and a recall of 0.15. Half of the falsely classified ranges are screams that are very close to shouts.

---

[6] Cf. http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html

[7] Cf. http://www.nist.gov/speech/tools/addttp4-11tarZ.html

4.1.12 Music Detection

The module detects music in the audio signal. The method we use was first proposed by [Minami *et al.* 1998]. Additionally, considering the method proposed in [Hawley 1993] we take stable power spectrum peaks as an indicator for music. An image-based approach is used to measure the presence of horizontal lines in the spectrum. The detection algorithm takes slices of a length of 10 seconds and calculates the power spectrum for 371 frequency bands up to 4kHz. A strong horizontal blurring operator is applied to the resulting spectral gray-scale image in order to emphasize horizontal lines and reduce other patterns (e.g., speech or noise). After an edge detection the image is binarized so that only horizontal lines longer than a certain threshold t_{length} are kept. For each time frame of a length of 0.5 seconds the sum of the edge pixels within this frame is considered to indicate music if it exceeds a defined threshold t_l. In addition, we use the distance of the sum to t_l as a degree of disturbance of the music. With $t_{length} = 16\ px$ and $t_l = 450\ px$ we achieve a precision of 0.95 and a recall of 0.87. This could be improved by including a beat detection algorithm as an additional feature.

4.1.13 Sound Event Detection

Our system implements a high-level sound detection method that can search the audio track of a movie for a number of previously learned sounds. We search for gunshots, explosions, crashes and screams in order to identify the movie's most dramatic and entertaining scenes.

Our approach is similar to the method proposed in [Hoiem *et al.* 2005]. For training, a small number of short example sounds between 0.5 and 2.5 seconds is cut from typical action movies and transformed into a simplified spectral representation with 17 frequency bands. A set of 63 descriptive features is then calculated from each sound. Among these features are the intensities in the different bands along with their standard deviation, a measure for the fluctuation of the overall intensity and rising or falling energy from start to end. The feature set is designed to be robust against differences in volume or length of the samples. The feature vectors from the positive and a great number of negative examples are then used to train a Support Vector Machine (SVM) for each sound type separately, using the LIBSVM[8] implementation.

In order to search a movie for any of the sound types, we traverse the movie in small steps of 0.1 seconds and calculate the above feature vectors over a length of 800ms (for gunshots), 1200ms (crashes and screams) to 2000ms (explosions). The compared length should roughly match the length of the training samples.

---

[8] Cf. http://www.csie.ntu.edu.tw/~cjlin/libsvm

The test movie performs with a precision of 0.4 and a recall rate of 0.4 for portions with gunshots. Problems arise if movies use sounds that are too different from the training examples, like futuristic weapons, or have very different audio characteristics in general (e.g., older movies in comparison to today's movies). The shrill nature of sounds is often mistaken for screaming by the SVM classification. Loud music is also a source for misclassification, as beats can be mistaken for gunshots or other types of crashing sounds. In general, the selection of training sounds has the greatest effect on the performance of the classification.

To improve the precision of the sound detection we use the output of other modules to filter out false positives. The sound volumes are used as a filter to count only loud enough sounds. Results from the music detection help to clean the list of detected gunshots from music beats. Explosion sounds will only be counted if they are accompanied by a sudden increase of brightness in the image histogram.

4.2 Generating Trailers of an Annotated Movie

After extracting the various features mentioned above, the second component of our system comes into play. The annotated movie containing the extracted features is used in combination with our semantic patterns in order to generate a trailer of the particular movie. [Zhu *et al.* 2003] uses a hierarchy for video summarization quite similar to that defined by our trailer rules base. However, it does not discuss video summarization for the movie trailer format. Also, they solely work with available video and audio footage. Our approach uses additional automatically generated animations and adds music and sound effects from footage-unrelated audio sources. The process that we propose of automatically generating trailers from annotated movies is split into the following sub-components:

1. Using a trailer rule base to create an abstract trailer structure that is used as a basis for
2. the selection of video and animation footage as well as music and sound effects to assemble a final trailer.

In Figure 3, the components of the generation process are displayed.

In more detail: In order to build a trailer we define a knowledge base that contains models for trailer structure elements and defines parameters for categories of video footage frame ranges. Before the generation process is started, we filter the movie annotation into the syntactic elements *clips* and classify them into *categories* with the mentioned parameters. Next, the trailer model is created based on rules in the knowledge base and influenced by the availability of footage and certain random events. In this way, the system generates a unique trailer model that is built to fit the available footage. The composition framework

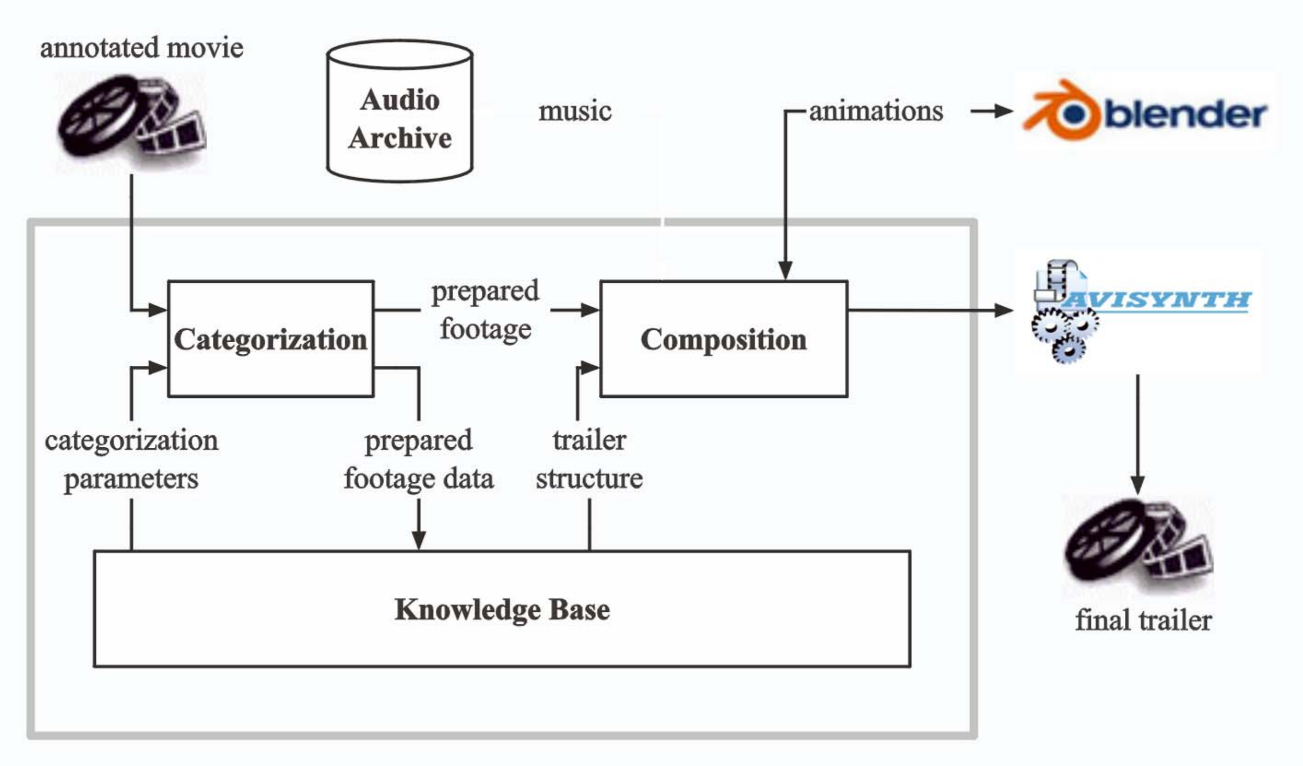


Figure 3: Diagram of the trailer generation process

translates the established trailer structure into specific trailer elements: Apart from video footage we incorporate runtime-created text animations for movie title, credits etc., as well as prepared music and sound effects content from our own audio archive. Footage, text animations, and audio are finally composed to a unique, fully automatically generated trailer based on trailer semantics.

4.2.1 Knowledge Base Functionality

In order to incorporate trailer semantics, we implement a knowledge base that is designed to hold the knowledge for trailer construction using the public domain software CLIPS[9]. The underlying knowledge of trailer syntax and semantics is modeled in an ontology. This trailer ontology includes classes for semantic structure elements (*patterns*) and syntactic elements (*clips*, *transitions*). We use relations to model our hierarchical view of the trailer structure with different types of patterns at the four upper levels and with different types of clips and transitions at the lowest level. The properties of clips (category, speed, volume, location) are implemented as slots and among these the category is implemented as a class of several slots again, specifying a list of video analysis attributes for its classification. The combination of several annotation attributes to a category leads to semantic higher-level knowledge about the footage.

---

[9] Cf. http://www.ghg.net/clips/CLIPS.html

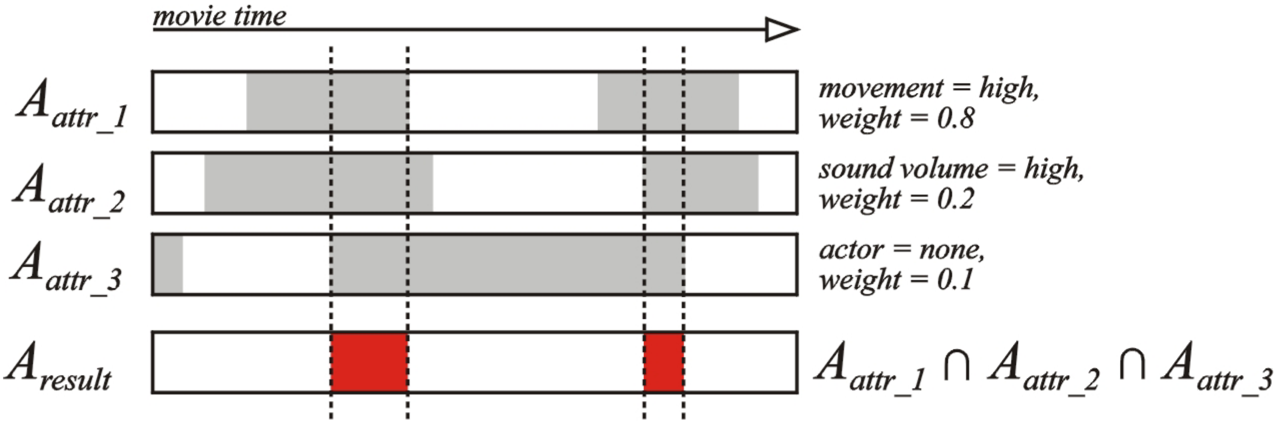


Figure 4: The intersection approach of the categorizer for sample footage

4.2.2 Categorization

Given a set of category definitions from the knowledge base the categorization module processes the annotation data in order to build clips for each category. We propose to build the clips based on frame ranges as opposed to a shot-based approach. As [Davis 1993] points out, a frame range approach allows the categorization process to be independent from scene/shot information and can provide categorized footage that starts or ends within a shot, or ranges over several shots. Hence, the main challenge is to determine frame ranges of the annotated movie, which match a certain category description from the knowledge base. Let A be a set of video frames and A_{movie} the set containing all original movie frames, then the frame set of an analyzed movie feature $A_{feature\ x}$ (e.g., all frames showing a face) is a subset of A_{movie}. The first step of the categorization process is to filter out the desired frames by corresponding thresholds (e.g., only get big faces indicating a close-up shot). This results in a new set which we refer to as an *attribute* frame set $A_{attribute\ x}$ with the following relations:

$$A_{attribute\ x} \subseteq A_{feature\ x} \subseteq A_{movie}$$

As illustrated in Figure 4, we process these attribute frame sets as tracks and perform an intersection of them. Furthermore, for each clip we calculate a probability value based on weighting factors assigned to the attributes. The result of the categorization process is a certain amount of footage clips for each category defined in our knowledge base.

4.2.3 Trailer Structure

Once the movie footage has been segmented and categorized, information about the amount of clips within each category is handed to the knowledge base. The system then builds the trailer structure on an abstract level. In order to introduce variety into the trailer models, each semantic element in our hierarchy has a selection choice of lower level elements assigned to it. These lists specifically define which items of a lower level a higher one may select. Thus, while offering multiple choices at each node in the trailer structure tree, the sequence of patterns can still be controlled to ensure consistency with the given trailer grammar. This approach grants easy and fast altering of the structure by linking more sub-patterns to a super-pattern or by deleting links. To avoid a purely random selection of a linked sub-pattern and to emphasize patterns that are more frequently used in movie trailers, a weighting system is attached to the selection logic. Based on the trailer ontology and on the availability of categories the knowledge base reasons about which parts of the trailer structure fulfill all requirements. In case of certain parts failing due to lack of footage, fallback structures are considered first. If no such fallback exists clip attributes are loosened: clips can then be chosen from random categories rather than specific ones. The result is a finished model of a trailer structure giving detailed information about which transitions to use, which background music to play, which clips of which category to show, what position within the movie they should come from, what speed they should be played at, and how high the volume of the original footage should be.

4.2.4 Selection of Clips

The footage clip objects in the trailer model come with properties regarding clip category, footage volume, speed, and location for footage selection. Footage clips are always selected from those matching the category of the clip object. Within this limitation, our system has three methods for clip selection:

1. Preferred location selection, based purely on the requested location and clip location in the movie, so the clip chosen is the one closest to the requested location.
2. Best clip from preferred location, which is similar to the preferred location selection with an addition of taking the quality of the clip into consideration so the clip chosen is the best clip available starting from the requested location.
3. A random clip of a given category is selected.

4.2.5 3D Text Animations and Audio

Text animations displaying information on movie title, release date, actor names, movie company as well as legal disclaimers are one distinctive feature of movie trailers and an essential component of a trailer structure. Our system uses the 3D software Blender[10], which offers animation and render control via Python scripts. Given a set of certain animation templates the composition system dynamically creates a script from which Blender produces one digital video file per animation ready to be used for final composition.

---

[10] Cf. http://blender.org

For additional music soundtrack and sound effects we provide the possibility to incorporate pre-produced sound files. Music files can be assigned to different trailer phases (that have to be predefined for a specific trailer pattern), while sound effects are divided into types. For every phase or element we have a choice of audio files.

4.2.6 Final Composition

Selected footage, animation clips and audio soundtrack are composed into a final video using Avisynth[11] scripts. Text animations are given sound effects to improve their effect. Changes in trailer soundtrack are masked by special transition sound effects. Fade/flash shot transitions (as determined by the trailer structure model) are implemented. The result is the finished trailer modeled according to a trailer structure created using our trailer ontology.

5 Automatic Trailers for Action Movies

Most trailers try to summarize the plot and setting of the announced movie and to introduce the relations between the main characters. Presently, the automatic extraction, analysis and generation of a narrative, or at least some kind of dramatic arc, seem hardly feasible. Therefore, our approach focuses on a genre that relies significantly more on visual sensation, speed and effects, than on narrative: the action movie. In order to generate trailers for this genre, specific grammar elements of action movie trailers need to be identified. In the next section we present parts of our sample pattern using a specific set of *sub-patterns*, *clips*, and *transitions*, which is based on a shot-by-shot analysis of various action movie trailers from the last 15 years.

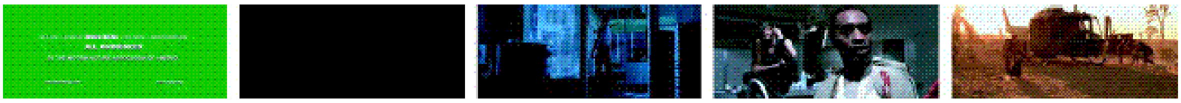
5.1 Applying the Rules to the Construction of a Trailer

We explicitly define 3 transition and 38 clip types, derived from 26 clip categories listed in Table 1. The definition of each category includes an appropriate set of attributes along with specific value ranges for the annotated features (t_{lo}, t_{hi}) and weighting factors (w_{attr}). One example of such a definition is shown in Table 2. An extension of our set of categories is possible and would be necessary to model and generate more complex trailers. On the second level of our hierarchy we identify five different phases (which are composed by a number of sub-patterns) as the basic structure in most action trailers. They are:

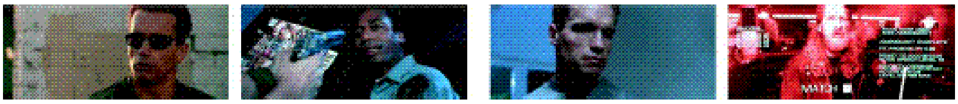
- Intro (slow and moody shots of locations and people together with speech establishing a conflict or introducing the main characters)

---

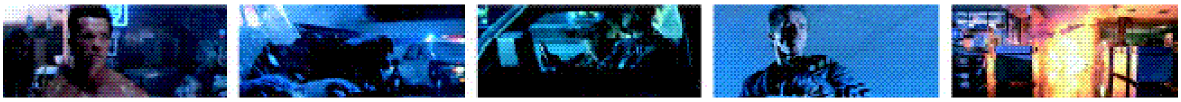
[11] Cf. http://www.avisynth.org



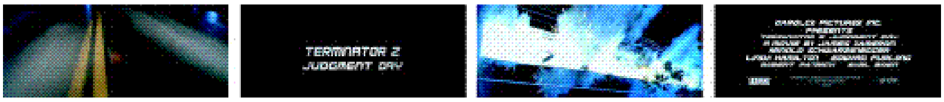
1 Greenscreen
2 Black
3 Setting
4 Setting
5 Setting



6 QuoteLong
7 Character1CUSilent
35 Character1Silent
36 SlowAction



37 PersonCUSilent
38 Gunshot
39 PersonSilent
40 PersonSilent
41 Explosion



53 QuoteLong
54 Title
55 Spectacular
56 Credits

Figure 5: 18 of 56 clips showing parts of our automatically generated Terminator 2 *trailer (complete Intro Phase: 1-6, middle part of the Action Phase: 35-41, and complete Outro Phase: 53-56). The corresponding type of category is given below each clip.*

- Story (medium fast shots of action and people together with dialogue to wrap up the task the main characters have to face)
- Break (a long and very significant or dramatic comment by one of the main characters - typically without background music)
- Action (a fast montage with loud sound of the fastest action scenes together with close-ups of the main characters)
- Outro (typically very calm or without any music and shows – sometimes mixed with close-ups or a short shot of one of the main characters uttering an extremely comic or tough comment – the title and credits of the movie together with a release date)

Left
Table 1: Clip Category and Phase Pattern relations in our sample Trailer Pattern (CU: close-up). I stands for Intro, S for Story, B for Break, A for Action, and O for Outro

	I	S	B	A	O
Transition					
FadeBlack	x	x			x
FlashWhite				x	
HardCut	x	x	x	x	x
Footage Clip					
Character1CUSilent	x	x		x	x
Character1CUSpeaking	x	x			
Character1Silent	x	x		x	x
Character1Speaking	x	x			
PersonCUSilent	x	x		x	x
PersonCUSpeaking	x	x			
PersonSilent	x	x		x	x
PersonSpeaking	x	x			
Quote	x	x	x		x
QuoteLong	x		x		x
Explosion		x		x	
Fire		x		x	
Gunshot				x	
FastAction				x	
SlowAction		x		x	
Spectacular				x	x
Shout		x		x	
Scream				x	
Setting	x	x			
Animation Clip					
ActorName				x	
CompanyName	x	x			
Credits					x
DirectorProducer		x			
Greenscreen	x				
Tagline	x	x			
Title					x

Attribute	t_{lo}	t_{hi}	w_{attr}
Movement	0.000	0.003	0.2
SoundVolume	0.0	0.1	0.1
Text	-1.0	0.0	0.1
Duration	60	500	0.1
CharacterFace	-1.0	0.0	0.3
CharacterSpeech	-1.0	0.0	0.1
MovieLocation	0.2	0.9	0.1

Table 2: Parameters for the Setting category

With the defined elements (clips, transitions and patterns), as well as their relations to each other we are finally able to describe a simple action movie trailer in a formal way. This description can be used by our system to generate an action trailer from any movie (as long as the automatic analysis provides enough footage for the different categories). In a simplified schematic way the relation between categories (being the basis for the clips), transitions and patterns that constitute our trailer structure can be described in a two-dimensional matrix as in Table 1.

5.2 3D Text Animations and Audio for Action Movie Trailers

In order to include text animations we provide our system with four animation templates which all have a different artistic style. Our audio archive is a collection of pre-produced sound files and consists currently of 37 music files and 22 sound effect files. Currently we use four categories of music files according to the mood of our trailer phases (Intro, Story, Action, Outro) and three sound effect types that are mostly used in professional trailers ("boom", "woosh" and "wooshbang").

6 Experimental Results

In order to evaluate the quality of our trailers, we asked 59 people to evaluate seven test trailers. For each of the trailers, the test people were asked to state whether they have seen the movie and to rate the same six aspects (with 1 as the lowest and 10 as the highest score). The test set comprised:

- Two professional trailers: *War of the Worlds* (Golden Trailer Award winner 2005) and *Miami Vice*
- One trailer for *The Transporter* produced by the video generation software muvee^{TM} (random shot selection)[12]
- Two trailers produced by our system with different levels of randomness: *Bad Boys* (random frame ranges), *Blade* (random *clip* selection)
- Two trailers produced by our system based on our Trailer Patterns: *Transporter 2* and *Terminator 2*

Except for the last trailer (*Terminator 2*, see Fig. 5), the test people did not know how the trailers were produced. After the impression of all trailers, the test people had the opportunity to watch any trailer again in case they wanted to adjust the ratings. The detailed scores of all trailers are shown in Table 3. As expected, the overall rating of the random trailers is significantly lower than any of the others, while *War Of The Worlds* performed best (see Fig. 6). The *Miami Vice* trailer that had been chosen as an example for a low-quality professional trailer was indeed given a bad score. Our own generated trailers reached an average score of 7.29 and 7.26, respectively. More than 80% of the viewers

---

[12] Cf. http://www.movee.com

	ss	*co*	*ce*	*ci*	*pi*	*av*
War of the Worlds	*8.41*	*7.91*	*7.79*	*7.47*	*7.40*	*8.16*
Miami Vice	*4.97*	*6.27*	*6.27*	*3.27*	*3.59*	*4.95*
The Transporter	*5.05*	*4.03*	*4.22*	*2.97*	*3.59*	*3.95*
Bad Boys	*4.64*	*3.67*	*3.41*	*3.19*	*3.22*	*3.52*
Blade	*4.16*	*3.24*	*3.24*	*4.07*	*4.07*	*3.43*
The Transporter 2	*7.47*	*7.54*	*7.90*	*6.80*	*6.80*	*7.37*
Terminator 2	*7.58*	*7.63*	*6.36*	*6.88*	*7.46*	*7.37*

Table 3: Detailed scores from the user testing; ss stands for scene selection, co for composition, ce for cuts & effects, ci for character introduction, pi for plot introduction, and av for advertisement value

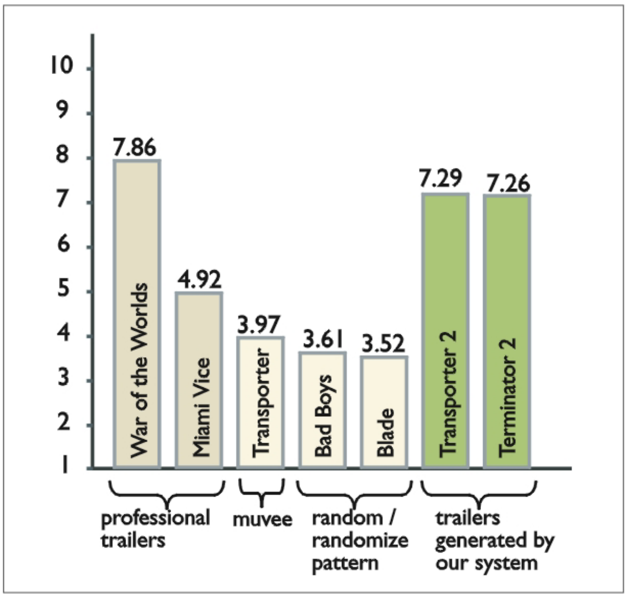


Figure 6: Mean score of the test trailers

rated them at score 7 or better. The question whether people had seen a movie or not seemed to have little or no impact on their judgment.

According to our survey, the weakest elements of our trailers are the introduction of main characters and topic / storyline with each around 6 points on average. This score is still good regarding the fact that we could not deliberately include those aspects into our trailers. A possible conclusion is that an illusion of storytelling and character introduction is created by extensively using quotes. At the end of the test screening, we asked the test viewers to rate the importance of six different aspects for Hollywood trailers. In average, people voted mostly for an even balance of the aspects. A slightly larger share was given to ‘dramaturgy’ (20.51%) and ‘action scenes’ (18.77%). The importance of ‘voice-over’ and ‘illusion of speed’ was rated a little lower (12.58%, 13.13%). In relation to this, we asked the viewers to rate the quality of integration in our *Terminator 2* trailer for the same aspects. The smallest share of votes was achieved for ‘voice-over’ (9.28%). This was to be expected since it is basically missing in any of our trailers. The best rates were given to ‘action scenes’ (22.26%), ‘music, animation and sound effects’ (18.6%) and ‘distinctive pieces of dialogs and statements’ (18.19%). This shows that our attempts of automatic categorization and composition appear to be successful in general. When interpreting the test results, the number of test people and the fact that they were not chosen representatively should be considered. However, the results suggest that our automatic trailers are in most respects comparable to original trailers and may even be more accepted by audience than low-quality professional trailers.

Sample trailers produced by our system can be downloaded from http://www.tzi.de/svp.

7 Conclusion and Future Work

This paper presents a novel approach of intensively using a knowledge base (rules) in combination with data automatically extracted out of a movie by different image and audio analysis techniques for generating a Hollywood-like movie trailer. First, the rules were defined which can be applied to various movie genres. Second, a system was implemented, which provides means for using extracted features to build a trailer according to any defined Trailer Pattern based on our trailer grammar. One such Trailer Pattern was created by manually analyzing several action movie trailers. Using our system we generated trailers for some action movies according to this pattern, and we evaluated our outcome.

After all, with our action movie trailers we proved that automatic trailer generation is not only possible, but can even achieve good results. This has been proven by tests we conducted with human subjects. Still, our trailers lack some elements a manually edited trailer comprises, e.g., telling a coherent story or voice-over narration.

The system can be improved by enhancing the modules extracting the data and by expanding the trailer grammar. Extensions of our modeled knowledge to incorporate editing knowledge for trailers of other movie genres, less standardized trailers or even other video summary formats are conceivable and can be achieved by adding more patterns, clips and transitions. Also, the classification of movie footage into semantic categories could be expanded by adding more categories (e.g., “kissing”, “fight”) based on more sophisticated image and audio processing techniques. Concerning the composition framework, improvements and extensions for the animation and audio inclusion are conceivable, notably to add more animation styles as templates and have a way of matching styles to movie content. Finally, the effect of a generated trailer can also be vastly improved by adding pre-produced generic voice-overs to the soundtrack.

A general different approach would be to manually add special information that might lead to more artistic trailers but would result in the loss of total automatism.

References

Arijon, D.: *Grammar of the Film Language*. Silman-James Press, 1991.

Babaguchi, N. & Kawai, Y. & Kitahashi, T.: Event Based Video Indexing by Intermodal Collaboration. In *Proc. of 1st Int. Workshop on Multimedia Intelligent Storage and Retrieval Management*, Orlando, 1999, 1–9.

Biatov, K. & Kohler, J.: An audio stream classification and optimal segmentation for multimedia applications. In *Proc. of 11th ACM Int. Conf. on Multimedia*, Fraunhofer Institute for Media Communication, Sankt Augustin, 2003, 211–214.

Chen, H.-W. & Kuo, J.-H. & Chu, W.-T. & Wu, J.-1.: Action movies segmentation and summarization based on tempo analysis. In *MIR ‘04: Proc. of 6th ACM SIGMM int.*

Workshop on Multimedia information retrieval, New York, NY. ACM Press, 2004, 251–258.

Christel, M. G. & Hauptmann, A. G. & Warmack, A. & Crosby, S. A.:. Adjustable Filmstrips and Skims as Abstractions for a Digital Video Library. In *ADL*, 1999, 98–104.

Davis, M.: Media streams: An iconic visual language for video annotation. *IEEE Symposium on Visual Language*, 1993, 196–202.

Hawley, M.: *Structure out of sound*. PhD thesis. MIT, Massachussetts, 1993.

Hoiem, D. & Ke, Y. & Sukthankar, R.: SOLAR: Sound Object Localization and Retrieval in Complex Audio Environments. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP*), 2005, vol. 5, 429–432.

Kernan, L.: *Coming Attractions – Reading American Movie Trailers*. Univ. of Texas Press, Austin, 2004.

Lienhart, R. & Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. In *IEEE ICIP*, 2002, vol. 1, 900–903.

Lienhart, R. & Pfeiffer, S. & Effelsberg, W.: Video Abstracting. *Communications of the ACM,* 1997 40(12):54-62.

Ma, Y. & Lu, L. & Zhang, H. & Li, M.: A User Attention Model for Video Summarization. In *Proceedings of ACM Multimedia*, 2002, 533–542.

Miene, A. & Dammeyer, A. & Hermes, T. & Herzog, O.: Advanced and Adapted Shot Boundary Detection. In Fellner, D. W. & Fuhr, N. & Witten, I. (eds.): *Proc. of ECDL WS Generalized Documents*, 2001, 39–43.

Minami, K. & Akutsu, A. & Hamada, H. & Tonomura, Y.: Video Handling with Music and Speech Detection. *IEEE MultiMedia*, 1998 5(3):17–25.

Ponceleon, D. & Dieberger, A.: Hierarchical brushing in a collection of video data. *Proc. of 34th Hawaii Int. Conf. on System Sciences*, 2001, 1654–1661.

Schrumpf, C. & Larson, M. & Eickeler, S.: Syllable-based Language Models for English Spoken Document Retrieval. In *Proc. of 7th Int. Workshop of the EU Network DELOS on Audio-Visual Content and Information Visualization in Digital Libraries*. Fraunhofer Institute for Media Communication, Sankt Augustin, 2005, 196–205.

Smith, M. & Kanade, T.: Video skimming and characterization through the combination of image and language understanding. *IEEE Int. Workshop on Content-Based Access of Image and Video Database*, 1998, 61–70.

Snoek, C. & Worring, M.: Multimodal Video Indexing: A Review of the State-of-the-Art. *Multimedia Tools and Applications,* 2005 25:5–35.

Uchihashi, S. & Foote, J. & Girhensohn, A. & Boreczky, J.: Video Manga: Generating Semantically Meaningful Video Summaries. *Proc. ACM Multimedia 99*, 1999, 383–392.

Wilkens, N.: Detektion von Videoframes mit Texteinblendungen in Echtzeit. Master's thesis, Universität Bremen, 2003.

Yambor, W. S.: Analyis of PCA-based and Fisher Discriminant-based Image Recognition Algorithms. Master's thesis, Colorado State University, 2000.

Yeung, M. & Yeo, B.: Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Trans CSVT,* 1997 7:771–785.

Zhang, H. & Kankanhalli, A. & Smoliar, S. W.: Automatic partitioning of full-motion video. *Multimedia Syst.*, 1993 1(1):10–28.

Zhu, X. & Fan, J. & Elmagamid, A.K. & Wu. X.: Hierarchical video content description and summarization using unified semantic and visual similarity. *Multimedia Syst.*, 2003 9(1):31–53.

[Table of Content]

Jörg R. J. Schirra

Conclusive Notes on Computational Picture Morphology

Abstract

As the thematic issue of IMAGE on computational image morphology attempts in particular to mediate between computational visualistics and other disciplines investigating pictures and their uses, the following remarks broaden the perspective again and relate the computational argumentations of the preceding papers to the more general discussion of image science. The two fundamental categories of picture syntax, the geometric base structure and the marker value dimension, are described. They are applied to the questions whether pictures with ill-formed syntax may exist at all, and if so, whether computers can deal with them as well. The overview finally extends the discussion to the limits of pictorial syntax studies.

As the thematic issue of IMAGE on computational image morphology attempts in particular to mediate between computational visualistics and other disciplines investigating pictures and their uses, the following remarks broaden the perspective again and relate the computational argumentations of the preceding papers to the more general discussion of image science. The overview also extends the discussion to the limits of pictorial syntax or morphology.

1 The Two Sides of Picture Morphology

In her influential book on picture syntax (1990), Fernande Saint-Martin distinguishes two kinds of properties of syntacto-morphological elements of pictorial signs that are often interpreted in the following manner (cf., e.g., Dölling 1999): *plastic* properties belong to the “material” of the picture vehicle while other properties are of a *perceptual-visual* nature, which means they are essentially “in the beholder’s eye”, constructed following the principles of visual perception and particularly Gestalt theory (cf., e.g., [Metzger 1966]). The geometric forms and their topological relations are given as typical examples for the latter, whereas color and texture are considered to be properties of the material as such. A combination of plastic attributes and visual-perceptive attributes forms Saint-Martin's version of a pixeme called "coloreme".[1]

In the papers presented in this volume, a related yet different distinction can be found at the basis of picture morphology in computational visualistics: the distinction of *geometric base structure* and *visual marker values*. In the essence, they form two different (classes of) abstract data types that have to be coordinated in order to form the logical manifold of the pixeme structure on which the algorithms described work.

Let us recall that – from the application point of view on computer science – an abstract data type is a formal version of one of the fields of concepts that structure the argumentations in the application domain in question: a calculus that covers the essence of the way people of the application domain speak about a certain phenomenon (cf. the introduction of this volume). In our case, the manner of speaking of picture researchers about color, texture, geometric forms, and their spatial relations is the reference to which solutions in computer science are rated.

The distinction between geometry and color is not – at least not primarily – one between something belonging (objectively) to the picture vehicle's material *versus* something constructed (subjectively) by the mechanisms of (visual) perception.[2] However, we indeed can talk about colors and the dependencies between them on the one hand, and about spatial entities and their geometrical or topological relations on the other hand without necessarily mixing the two threads of argumentations. They can be treated as independent, and thus can be considered as being governed by – *prima vista* – autonomous fields of concepts. Thus when dealing with picture morphology, computational visualists ought to consider, first, one set of abstract data types covering the logic of color, and another set of types describing the logic of space. They, then, have to combine one data type of each of the two groups in order to gain a calculus (more or less) equivalent to the argumentations concerning pixemes (cf. Figure 1).

1.1 The Geometric Base Structures: The Logic of Locational Gestalts

Pictorial syntax deals, coarsely speaking, with the limited, spatial arrangement in two dimensions of visual distinctions (or, for short: of colors). The logic of spatial arrangement in

---

[1] The difference between the conception of pixemes and Saint-Martin's coloremes becomes clear in the following determination: "[A coloreme] corresponds to that aggregate of visual variables perceived in the visual representation by the way of an ocular fixation, or focus of the gaze. … A coloreme is defined […] as the zone of the visual linguistic field correlated to a centration of the eye. It is constituted by a mass of energetic matter presenting a given set of visual variables." (Saint-Martin, 1990, 5). Saint-Martin's determination of coloremes concentrates on psychophysical aspects leaving the abstract formal dimensions of geometrical form/position and color/texture implicit.

[2] It is, after all, quite a strange idea to attribute color – classically treated as the paragon for secondary (i.e., subjective) properties – to the material and not to our perceptive apparatus. Color constancy depends more on a relatively complicated neuronal mechanism then on object properties.

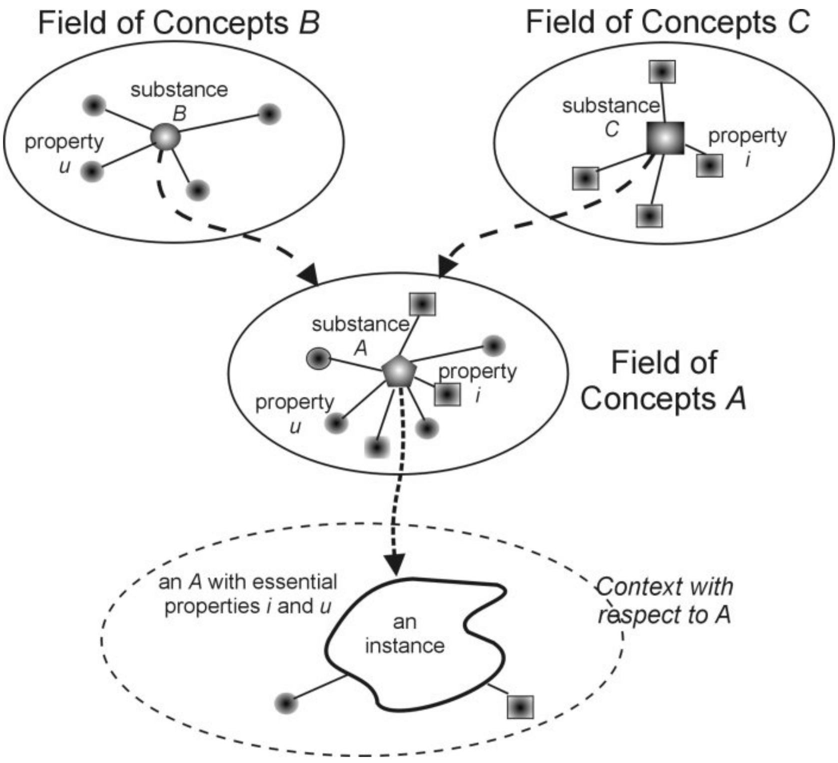


Figure 1: Combination of fields of concepts

The combination of one field B (e.g., the field of geometric concepts) with another one C (for instance the one of color concepts) explains the structure of a more complex field of concepts A regulating instances with coordinated properties from the two constituting fields (for example concepts governing colored geometric entities, i.e., pixemes)

general is covered in the essence by the various calculi of geometry (cf., e.g., [Aiello *et al.* 2007]). Abstract as those calculi are, they formalize central aspects of our concrete interactions with any kind of pure spatial configuration, and can be more or less immediately translated to abstract data types. Correspondingly, all the papers in this volume refer to one or the other of such a geometric base structure in two dimensions.[3]

The locational organization of pixemes is in fact not perceivable as such.[4] Like the temporal base structure of music that can only be perceived as organizing a sequence of distinct auditory markers – difference of pitch or harmonic progression, change of volume or variation of timbre – the perception of the spatial base structure of pictures depends on visible differences: visual markers usually subsumed under the expression “color”. Indeed, color in this general sense includes hue, saturation and intensity as well as texture or even homogeneous temporal variations thereof. It is exactly the change of any one of those values that induces the border of a pixeme, and thus determines the spatial “rhythms” of the picture. Although underlying most of the papers in this volume, only few of the authors have

---

[3] Time may occasionally be added to the base structure as an additional “spatial” dimension.

[4] Space (and time) is, using the words of I. Kant, not an empirical phenomenon but a transcendent category used by perception to organize empirical phenomena (CpR).

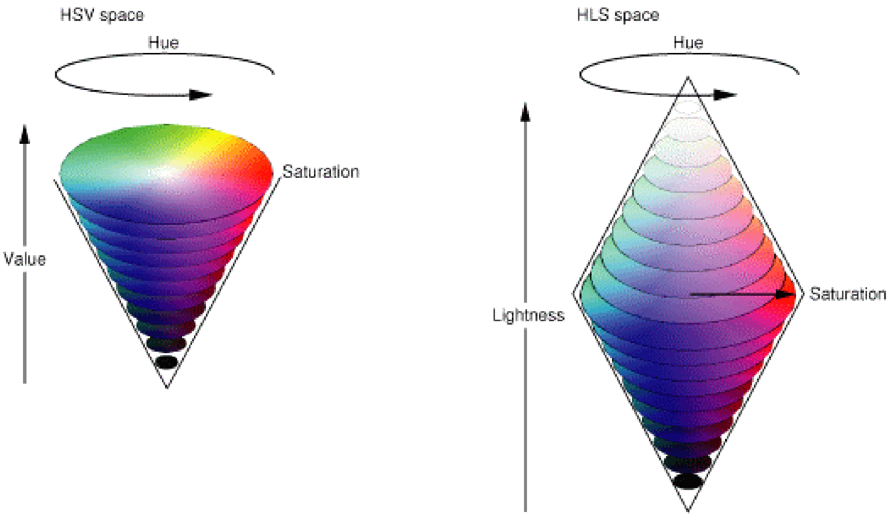


Figure 2: Graphical-geometrical presentation of two color models: neighborhood and transpositions with respect to the axes or symmetry centers are equivalent to relevant relations between the corresponding color concepts

elaborated this aspect in some details; they mostly rely on the everyday knowledge about this dimension. However, color theory is often not as simple, and some additional remarks about the visual marker system should be added in this conclusion.

1.2 The Visual Marker Values: Color and Texture, Reflection and Transparency

The various systems to formally cover color (in the closer sense) in computer science are indeed, we assume, well known: Every painting program or computer system for picture manipulation offers at least RGB or HSV. Those “color models” are essentially equivalent to each other and do not need a detailed description at this place. They basically implement the system of color concepts we normally apply when speaking about colors and the dependencies between them (cf. Fig. 2).

We should note that we meet once more with color the problem of formalizing a seemingly dense dimension: Between any two colors there appear to be more colors. And again, we depend on a perceptual system with a limited resolution in color distinction.[5] In contrast to locative resolution however, there is no such thing in “color space” as a natural “zooming operation”: the members of some pairs of colors are only distinguishable by means of a complicated technical device like spectral analysis that has no equivalent in non-technical human behavior.[6] We may therefore take color without real simplification as a syntactically discrete dimension with a resolution just below the threshold of human perception. Correspondingly, contemporary computer systems offer a data type for homogeneous colors

---

[5] Are there arguments for taking color space to be even continuous? Physics at least assumes a continuous spectrum (range of wavelengths or frequencies) of electromagnetic waves implementing color, though the relevance of this conception for color perception is only quite indirect.

[6] Zooming locational resolution by microscopes or telescopes can be viewed as a technical equivalent to concretely approaching the scene perceived, as was already indicated in the introduction.

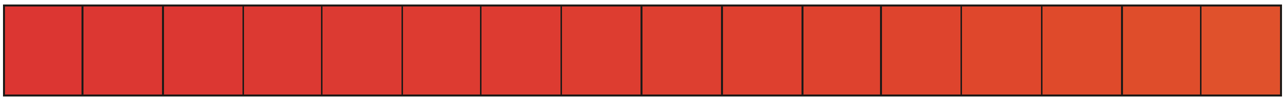


Figure 3: The same color?
Starting from RGB (255,0,0) in the left box, the color in each box is changed by (0,5,0) till (255,75,0): that is, “between” two adjacent boxes, four more color values are possible in RGB, while most humans cannot distinguish the colors in two adjacent boxes

with more than 16.5 million values together with methods to select and manipulate them easily along the dimensions of our color concepts. Two immediately neighboring color values of that system are for most humans indistinguishable (cf. Fig. 3).[7]

Homogeneous color as covered by the color models mentioned above is the central aspect of the visual marker dimension, but not its only aspect. More often, the visual markers are given as fine-grained textures that only appear as more or less homogeneous if the spatial resolution is not too high. In these cases, zooming reveals that a locale distribution of homogeneous colors is in fact relevant (or even fields with textures on a still finer level). However besides the zooming, textures are perceived, remembered, and even imagined not as a particular spatial distribution of individual (homogeneous) color values but as a different kind of visual marker values more or less analogous to accords in music (with tones as analogon to colors). The system of visual markers consists in fact of – at least – two levels. Although the two levels are not completely independent from each other, they follow quite different internal rules.[8]

1.3 Contextual Aspects of Picture Morphology

Finally, by means of transparency and reflectivity something distantly related to deictic elements in verbal signs is included in the visual marker dimension, as well. Those two phenomena of color in the broad sense are seldom dealt with in computational picture morphology. Recall as examples of corresponding traditional pictures stained church glasses or Mexican or Turkish folk art with build-in pieces of mirrors, or see Figure 4.

Note that the effects of reflectivity and transparency in the examples cannot be ascribed to the picture vehicle as such – it has to be considered in (and in contrast to) changing situational contexts. In every single context (i.e., arrangement of objects and lights around the image), the transparent and reflective regions of the picture have a fixed appearance indis-

---

[7] Moreover, there are few technical devices that really reproduce each single value distinctly.

[8] As textures can technically be reduced to fine-grained patterns of homogeneous colors, the most common way to deal with them in computational visualistics is by using a sample. More ambitious analytic solutions for a corresponding data type concentrate on characteristic structural, statistical or spectral parameters [Long *et al.* 2000]. Structural parameters characterize textures according to geometric relations between corresponding homogeneous sub regions while statistical texture parameters measure the locale variations of visual qualities (e.g., granularity, regularity, line-likeness): the feature “roughness”, for example, depends on the fractal dimension of the intensity variations relative to spatial displacement (cf., e.g., [Wu & Chen 1992]). For spectral approaches, the Fourier transform of the texture is calculated as the basis of further analyses.

Figure 4: A Rather Extreme Example for Reflectivity and Transparency as Aspects of the Pictorial Marker Value Dimension (Toby Mason, Forming of the World*, 1997)*

tinguishable in that respect from other regions – they may have been marked by homogeneous colors or textures just as well (as in fact in Figure 4). An observer perceives regions as being transparent or reflective only if changes in the context do indeed modify the distribution of marker values, and hence the arrangement of pixemes. The phenomenon is also directly important for computational visualists when combining pictures in layout (mostly transparency) or 3D graphics (transparency and reflection). Of course, an adequate conceptualization in the data type »image« must explicitly include such “indexical marker values”; in general, we cannot replace them by one arbitrarily induced distribution of homogeneous colors or textures.[9]

There also exists a contextual factor that influences the geometric base structure: While the calculi of, for example, pure mereogeometries only provide symmetric spatial dimensions, gravity – or the up-down polarity induced by it in the perceptual system of the observer – introduces an asymmetry in the spatial arrangement of the pixemes. However, like the quasi-indexical elements of reflectivity and transparency, the influence of gravity

---

[9] As a standard for transparency, an additional dimension of marker values – beside hue, saturation and lightness (or the other dimensions of color in the close sense used equivalently) called the “alpha channel” is regularly used in computational visualistics. Obviously, this “transparency can only be employed internally and does usually not extend to the external presentations of the picture: Obviously, a paper printout does not become transparent accordingly. Reflectivity also poses some particular problems in computational visualistics, as the reflection of the observer needs to be dealt with as well – a pragmatic problem that cannot be solved in an easy manner.

may be taken not as a syntactical aspect of pictures at all, but as an element of pictorial pragmatics.

In conclusion: the formal treatment of pictures in computational visualistics covering the syntactic aspects rests essentially on two basic data types and their interaction: first, the base structure of position and form, for which the calculi of mereogeometry are quite promising general candidates at present; second, the field of marker values based on a discretized range of homogeneous colors and an additional dimension for transparency (and perhaps reflectivity), offering further structural principles for the secondary level of textures.

2 Do Syntactically Ill-Formed Pictures Exist?

Let us consider as a final aspect of pictorial syntax a thesis that is often discussed in general visualistics: in contrast to verbal expressions that have syntactically ill-formed counterparts, there seems to be no such thing as a syntactically ill-formed picture (cf. [Plümacher 1999]). Whereas, for example, the syntactic structure of a verbal language may be described by just *one* Chomsky grammar, so that expressions not described by that grammar are considered ill-formed (with respect to that grammar), *any* expression in *any* two-dimensional visual L-system, *any* mereogeometric configuration associated accordingly with marker values, *any* flat surface makes, it seems, a picture vehicle. The reason appears to be essentially that the geometric basis of pixemes is dense, and *any* potential combination of pixemes can be used as a picture.

2.1 What Are Morphologically Ill-Formed Pictures

The distinction between the dimension of the geometric base structure and the dimension of the marker values is indeed quite helpful to understand the difference between verbal sign systems and pictures, also with respect to ill-formedness. Indeed, those discussing this issue do usually not mention *damaged* screens: Cuts, holes, and burned regions disrupt the homogeneous topology that is part of the pictorial base structure. Cuts, for example, separate neighboring pixels: are they neighboring anymore or not, we cannot really say (cf. Fig. 5). Suddenly, there is non-space in picture space – which is certainly not equivalent to fully transparent regions. After all, a cut in a “Rembrandt” results not just in another picture but in a destroyed picture. So, our counter-thesis is that pictures might quite well be counted as syntactically ill-formed if the underlying geometric structure is disrupted.

As with syntactically ill-formed verbal expressions, which may nevertheless be used efficiently for communication, syntactic well-formedness is no necessary criterion for a picture to be employed: A certain art form in the middle of the 20th century, particularly exemplified

Figure 5: An ill-formed Picture? (A. E. Arkhipov: Peasant Girl *(1920s), with a tear)*

Figure 6: Lucio Fontana: Concetto Spaziale *(1965)*
Intentionally cut screen said to refer to the "materiality" (i.e., the syntax) of pictures

by L. Fontana (1899–1968), plays exactly with this syntactic deviation from well-formed images. Fontana's "cut pictures" are reflective pictures that focus our attention on the "materiality" – or in our terminology: on the geometric base structure – of pictures exactly by means of the violation of that very basis (cf. Fig. 6, [Whitfield, 1999] and [Sachs-Hombach 2002, 164f.]).

Syntactic disorders – much like reflection or transparency – have a deictic quality: viewed from merely one perspective, a cut may not be noticed for what it actually is, but taken as another (in all probability semantically strange) pixeme. Only the movement of the beholder makes clear that the spatial tissue of the syntactic base structure itself has been broken.[10]

Let us, before turning to the question of how such intentionally ill-formed pictures might be dealt with in computational visualistics, shortly look at the purpose of such pictures. Employing syntactically ill-formed picture vehicles on purpose is mostly restricted to art. More precisely, these pictures are associated with a special mode of use, as the communicative act they are used for deals with the pictorial sign act itself, and hence, among other aspects, with its syntactic structure.

---

[10] Similarly, "blind spots" on a TV screen – i.e., locations where due to some technical problem no light is emitted – can only be recognized as such if the picture moves accordingly, so that a change in the marker values at those spots had to occur. As that is not possible, the location can be identified as not being part of the base structure, which thus must have a geometrical anomaly.

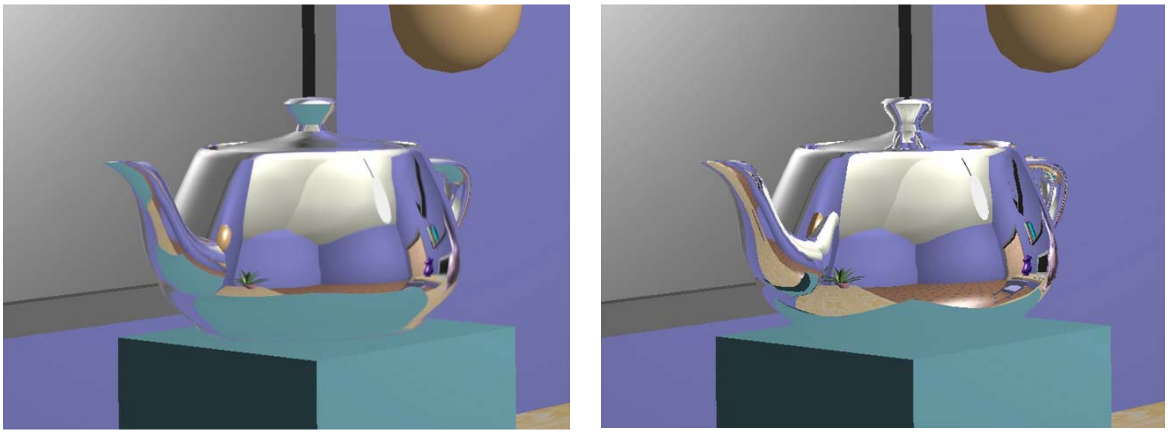


Figure 7: Exemplifications of the Reflections of Depicted Objects Reached by the Computer Graphics Algorithms 'Environment Mapping' (left) and 'Ray Tracing' (right) Using the Notorious "Utah Teapot"

2.2 The Reflective Use of Pictures

Pictures that are not used in the primary sense of showing their content but instead of demonstrating aspects of pictorial communication are usually called reflective pictures. Many pictures of art indeed are reflective pictures. They are called 'reflective' as they are used to communicate pictorially about the conditions of picture uses and picture productions, or for short: about picture communication and its constituents itself.[11]

Reflective pictures differ from other pictures by a different attitude of the beholders.[12] In this attitude, we show ourselves a picture as an example of one or the other of the many aspects of pictorial communication. Indeed, this is what we usually do when visiting an art museum, and pictures of art can generally be interpreted as pictures that are made specially for being received in reflective mode. In consequence, distinguishing reflective pictures from others is an aspect of pragmatics rather than syntax. However, as any aspect of picture use may be focused on when using a picture reflectively, syntactic features also play a role occasionally.[13]

---

[11] The scope of reflection may indeed reach far beyond morphology: from exemplifying the ability of the picture maker to produce highly deceptive pictures (as plays a major role for many *nature morte* of the 16th century) to the pictorial critique of the focus on naturalism. The central theme of the American art style 'photorealism' of the 1960s and '70s, for example, is an indirect critique of the visual access to reality in the modern industrial societies: an access that is almost totally mediated by technical reproductions, and thus open to all kinds of hidden manipulations [Held 1975]. The images of artists like Close, Bell, and Morley do not try to show reality in a photo-like realism; their subject is more precisely the mediated access to what is believed to be reality by media that are assumed to present subjects naturalistically.

[12] This special attitude has been called 'the reflective mode of reception'; cf. Schirra 2005, Sect. 3.5.1 and 4.4.5.

[13] In pictures of art, the „eigen values" of the picture, i.e., its syntactic features brought up in its reflective use, may dominate the semantic "depiction values" or even completely supplant the latter as in abstract art; cf. Buchholz 1999, 256f.

Reflective picture uses are not restricted to the artistic contexts: Quoting a picture is basically showing a picture vehicle in the reflective mode, too. Analogous to the use of verbal quotation, for instance of some example sentences in a linguistic textbook, which must not be confounded with the normal (direct) uses of those sentences, the application conditions of a quoted picture are quite different from its direct use: e.g., showing a Renaissance sacral picture in a university seminar on art history vs. using it for prayer in a chapel.

2.3 Reflectively-Used and Ill-Formed Pictures in Computational Visualistics

Although reflective pictures of the kinds used and invented in art are seldom relevant in computational visualistics, at least the particular use conditions of example images employed in texts on pictures may be considered important. We may quote pictures in order to exemplify a certain algorithm of computer graphics (cf. Fig. 7) or image processing (cf. Fig. 8).

Therefore, an aspect of picture production (hence use) is communicated by means of the presentation of such a picture; what is to be seen (as those pictures are usually of the representational kind) is more or less contingent. The frequency of teapots in pictures presented in computer graphics books does by no means communicate a particular addiction to the beverage or the receptacle, nor is the fact that a horse is presented pictorially important for the original use of Figure 8.[14] How the object chosen is depicted, how the visual Gestalt relates to the object, and in particular: how that relation again is linked with some aspects of the algorithm exemplified, that is what the sender of such a pictorial message normally intends – and what the receivers expect to be told in those communicative circumstances. Those pictures are therefore clear cases of reflective pictures, as well. In particular those pictures exemplifying, like Figure 8, segmentation algorithms are indeed quite important for the discussion of pictorial syntax in computational visualistics: Those algorithms operationalize the concepts of pixeme formation.

As the reflective use of a picture determines in fact a pragmatic category, reflective pictures are usually not distinguished syntactically from other pictures: They correspondingly are not dealt with in a special way in the computer as long as their syntactic characterization alone is considered. In more complex applications, like interactive systems that have to consider semantic and pragmatic aspects at least to a certain degree, the reflective use has not been employed so far.[15]

---

[14] Indeed, Figures 7 and 8 are thus already quotations of pictorial quotations – the reflective use of reflectively used pictures.

[15] Apart from quoted pictures used reflectively to refer to syntactic properties and algorithms associated to morphology, syntactic aspects are at least sometimes reflected by artistic computer pictures. Huber [1997, 188] mentions in a survey on web art, for instance, a piece of John Simon jr. that certainly evokes in the beholder the discussion on syntactic properties of pictures: "*In a second work for the web from John Simon jr. titled 'Every Icon' (http://stadiumweb.com/EveryIcon), a Java applet generates all combinations of black and*

Figure 8: Exemplification of a Particular Segmentation Algorithm (Segmentation by Aggregated Weighting): Input Image and Depictions of Results for Three Parameter Settings

Ill-formed pictures, on the other hand, are syntactically special, and hence, it seems, should be dealt with in computational picture morphology accordingly. First, however, it is important to distinguish the digital pictures of ill-formed real pictures from the ill-formed digital pictures. Certainly, the digital photos of a Russian painting with a large tear or of a work of Fontana – see again Figures 5 and 6 – are syntactically *not* ill-formed as well. They are quite regular pictures with an undisturbed spatial base structure, with some of their pixemes marking the regions of the tear or cuts, just as other pixemes in other pictures mark the regions of open doors or other holes of an object depicted.[16]

If the base structure for pictorial syntax is given by some calculus of geometry, any inconsistent geometrical description can be counted as the computational analogon of a damaged screen. Thus, picture files that have been incompletely transferred from a digital camera or from some Internet server could indeed be taken accordingly. However, presenting them in the visual form – projected on a screen or printed on some paper – the missing spatial coherence is not realized but substituted as in the case of the photos of an ill-formed picture. Thus, although syntactically ill-formed pictures theoretically exist in computational visualistics, as well, they are, at least up to now, not practically accessible. That is: up to now, it is not possible to adequately "computerize" one of the "Spatial concept" pictures of Fontana with their characteristic cuts.

---

white squares. The work runs since March 1, 1997. It can be viewed only in its beginning – a computer has to run with that little application day and night for years". In fact, about several hundred trillion years are necessary, Simon points out on the commenting web page, for the program to generate systematically all variations of the 32 * 32 pixel matrix used on the way from completely white to completely black.

[16] The deictic quality of the spatial disruption is, then, also lost.

Figure 9: Example of Watermarking
From Left to Right: Original, Watermarked Original, and Watermark Image Used (i.e., the difference between the other two pictures)

3 The Limitations of Picture Morphology

There still remain many aspects of pictorial morphology not covered in this volume. One particular question is the one of the identity of pictures – a question that in Western culture has mainly been answered in close connection to the evolving focus on genial artists by referring to the identity of the corresponding picture vehicle, the "original".[17] Other cultures have developed different conceptions that are more closely linked to the relation between a piece of music and its individual performances.[18] The generic concept of pictures encompasses a sub concept where the picture is fixedly bound to a certain individual picture vehicle, as well as a sub concept with an elaborate two-level conception.[19]

Since instances of the data types for picture vehicles cannot *per se* be seen but have to be made visible by means of a computer screen, beamer or printer, the computational treatment of pictures favors the second type. The concrete instantiation of the picture vehicle may thus differ more or less slightly. This principal morphological "slackness" is even used for certain syntactic solutions to pragmatic problems: By means of "watermarking" a picture vehicle, i.e., subtly modifying the morphology, the authenticity of a picture is ensured, and uses hurting copyrights can be verified (cf. Fig. 9).[20]

---

[17] It may in fact be a good hypothesis that such a conception of »image« is directly associated to Goodman's conclusion mentioned in the introduction: that proper copies can only be made from signs that are not syntactically dense.

[18] In many tribes of Australian and American indigenous people, a frequent means of cultural expression are sand drawings. Such pictures are "drawn" by strewing colored sand in patterns on a relatively flat part of the floor, or by pushing lines and dots with a stick or the fingers in flat monochrome sand or mud. They are usually produced in the course of a religious ceremony, which also requires the picture being destroyed at the end. Yet the pictures are said to be the same in different actualizations of the ceremony; cf. [Morphy & Smith Boles 1999].

[19] During the 20th century, the later sub concept has become more important in Western art, in particular with the employment of corresponding technical tools like video or the computer by the artists.

[20] When the expression 'authenticity' is currently used in computer science, it does not refer as usual to the coordination between a sender's attitude and the message's content; that relation is usually not accessible for the systems. There are, however, commitments of the computer as a medium (or rather, of those providing the medium), among them the commitments called 'integrity' and 'authenticity'. Integrity is granted if the receiver of a message gets exactly what the sender has sent, i.e., nothing has been left out, added or changed. Authenticity in the technical sense means that the receivers can be sure that the apparent sender

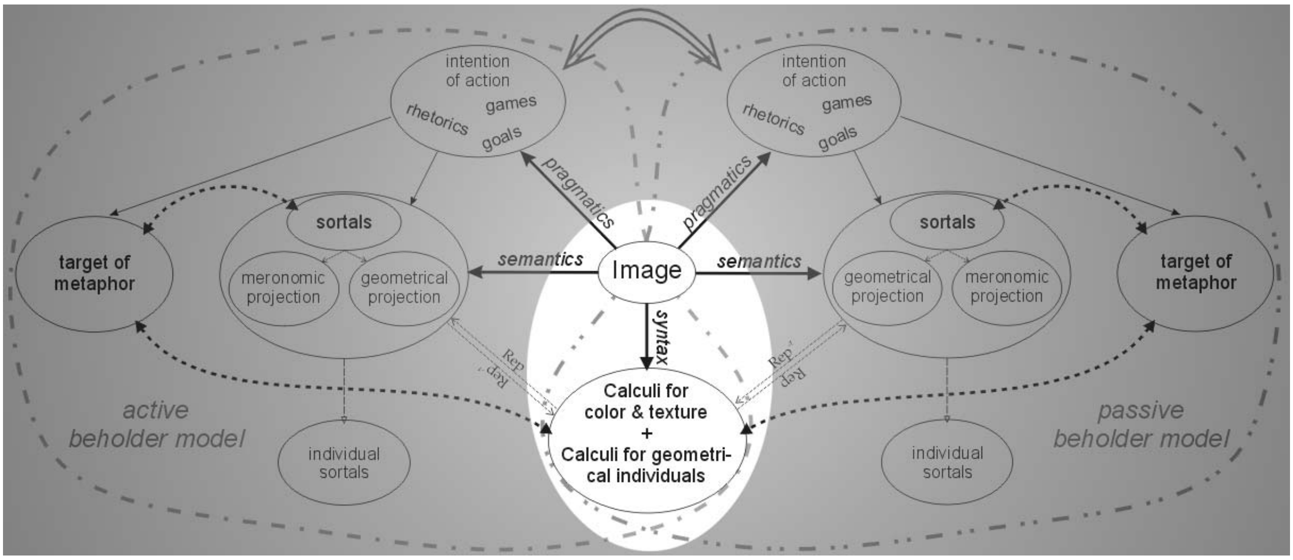


Figure 10: The Spot Light is on Pictorial Syntax –
but there is a lot more of (computational) picture theory

Providing structures isomorphic to the morphological characteristics of images may indeed be sufficient for handling pictures by means of a computer – after all that structure is (or rather would be) exactly equivalent to all the relevant aspects of the picture vehicles. However, computational visualistist should not be satisfied, as pictures are not merely picture vehicles but much more complicated entities. Not everything flat and covered with regions of textures is already a picture. With its pictorial metaphor of a theatrical spot light, Figure 10, the basis of which was originally used in [Schirra 2005] as a coarse overview on a version of the complete data type around »image«, illustrates how small the syntactic range of questions is indeed compared with the other conceptual facets of pictures.

Many of the papers collected in this thematic issue of IMAGE refer to semantic and pragmatic aspects, in the strict structural considerations as well as in the practical applications, since the syntactic problems they investigate only make sense in the context of those features and cannot be solved in isolation. Even the syntactic grouping of pixemes into entities of higher order takes into account not only the morphological attributes of the corresponding elements but also more or less every other pixeme present in the picture: The grouping is highly context-sensitive. Indeed, the identification of the pixemes particularly in a figurative picture depends to a high degree on the picture's content, i.e., what is depicted.

If we – the computational visualists – do not also consider the particular contexts of use that make us take a flat object for a picture, there is no way to, for example, select rationally from a given set of pictures the one to be best presented to a certain computer user

---

is the real one. Signatures are a common means of authentification (e.g., of letters, works of art). In combination, the two commitments also guarantee that the sender cannot deny to have sent the message in question (i.e., 'non repudiation' of the message) – a feature with important juridical implications, too.

under some specific conditions at hand. An overview on computational picture morphology cannot deal with the multitude of other questions associated with the concept (or data type) »image«. But – apart from explaining for those not too familiar with computer science our insights into the syntactic aspects of pictures and the options and restrictions of the computational approaches – it may help us to see the limitations of syntax, and to better understand the demands, the other image sciences express to computational visualists.

Literature

Aiello, Marco & Pratt-Hartmann, Ian & van Benthem, Johan: What is Spatial Logic? In: *dito* (eds.): Handbook of Spatial Logics, Berlin: Springer, 2007, 1–12.

Buchholz, Kai: Zum Verhältnis von Bildsyntax und Darstellungswert am Beispiel künstlerischer Grafik. In: [Sachs-Hombach & Rehkämper 1999, 255–270].

Dölling, Evelyn: Kategorialstruktur ikonischer Sprachen und Syntax der visuellen Sprache. In: [Sachs-Hombach & Rehkämper 1999, 123–134].

Held, Jutta: Visualisierter Agnostizismus. *Kritische Berichte,* 1975 3(5/6):1–7.

Huber, Hans-Dieter: Internet. In: Hirner, R. (ed.): *Vom Holzschnitt zum Internet. Die Kunst und die Geschichte der Bildmedien von 1450 bis heute.* Ostfildern-Ruit: Cantz, 1997, 186–189.

Long, Hui & Leow, K. W. & Chua, F.K. & Kee, F.: Perceptual Texture Space for Content-Based Image Retrieval. In: *Proc. MMM*, 2000, 167–180.

Morphy, Howard & Smith B. M. (eds.): *Art from the Land: Dialogues with the Kluge-Ruhe Collection of Australian Aboriginal Art*. Charlottesville: University of Virginia, 1999.

Plümacher, Martina: Wohlgeformtheitsbedingungen für Bilder? In: [Sachs-Hombach & Rehkämper 1999, 47–56].

Sachs-Hombach, Klaus: *Elemente einer allgemeinen Bildwissenschaft*. Habilitationsschrift, FGSE, Univ. Magdeburg, 2002 (revised version: Köln: Herbert von Harlem, 2003).

Sachs-Hombach, K. & Rehkämper, K. (eds.): *Bildgrammatik: Interdisziplinäre Forschungen zur Syntax bildlicher Darstellungsformen*. Magdeburg: Scriptum, 1999.

Saint-Martin, Fernande: *Semantics of Visual Language*. Bloomington: Indiana Univ. Press, 1990.

Schirra, Jörg R.J.: *Foundation of Computational Visualistics*. Wiesbaden: DUV, 2005.

Sung C.-K.: Extraktion von typischen und komplexen Vorgängen aus einer langen Bildfolge einer Verkehrsszene. In: Bunke H. & Kübler, O. & Stucki, P.: (eds.): *Mustererkennung 88*, Berlin: Springer, 1988, 90–96.

Whitfield, S.: *Lucio Fontana*. London: Hayward Gallery Publishing, 1999.

Wu, C.-M. & Chen, Y.-C.: Statistical Feature Matrix for Texture Analysis. *CVGIP: Graphical Models and Image Processing*, 1992 54(5): 407–419.

Sources of Pictures

Fig. 1 quoted from [Schirra 2005], p. 123.

Fig. 2 quoted from Apple Developer Connection: “About Color Spaces” → http://developer.apple.com/documentation/Cocoa/Conceptual/DrawColor/Concepts/AboutColorSpaces.html (as of June 2007).

Fig. 4 quoted from Toby Mason: Reflective Glass Mosaics – Forming the World, 1997 → http://users.erols.com/masont/artpages/forming.html (as of June 2007).

Fig. 5 quoted from Olga Nikolic-Litwin: Patina Studio Conservation of Paintings and Icons → http://www.patinapal.com (as of June 2007).

Fig. 6 quoted from Maria Hynes: Rethinking Reductionism. → http://culturemachine.tees.ac.uk /Cmach/Backissues/j007/art_res.htm (as of June 2007).

Fig. 7 quoted from Watt, Alan & Policarpo, F. (2001): 3D Games: Real-Time Rendering and Software Technology. New York: Addison Wesley, Fig. 7.25.

Fig. 8 quoted from Brown University, Amir Tamrakar: Project 5: Comparison of various region-based segmentation algorithms → http://www.lems.brown.edu/vision /courses/computer-vision/ (as of June 2007).

Fig. 9 quoted from Peter Meerwald: Digital Watermarking → http://www.cosy.sbg.ac.at /~pmeerw/Watermarking/ (as of June 2007)

Fig. 10 modified from [Schirra 2005], p. 265.

Contact Addresses:

Borgo, Stefano: LOA, ISTC-CNR, Trento (ITALY):
borgo@loa–cnr.it
du Buf, Hans: Univ. of Algarve, Campus de Gambelas, Faro (Portugal):
dubuf@ualg.pt
Engelhardt, Yuri: Univ. of Amsterdam (The Netherlands):
engelhardt@uva.nl
Hermes, Thorsten: SVP Group, Technologie-Zentrum Informatik, Bremen (Germany):
hermes@tzi.de
Isenberg, Tobias: Univ. of Groningen (The Netherlands):
isenberg@cs.rug.nl
Kurth, Winfried: Brandenburg Univ. of Technology at Cottbus (Germany):
wk@Informatik.tu–cottbus.de
Schirra, Jörg R.J.: Otto-von-Guericke Univ., Magdeburg (Germany):
joerg@jrjs.de

IMAGE
ZEITSCHRIFT FÜR INTERDISZIPLINÄRE BILDWISSENSCHAFT

Call for Papers

The Interactive Image: Pictorial Pragmatics in Computational Visualistics

Thematic issue of IMAGE – Journal on Interdisciplinary Image Science, 2010

Editor: PD Dr. Jörg R.J. Schirra

Apart from its widely-used methods for automatically dealing with pictures, computer science has contributed to the subject of image science as well in one particular aspect: it has provided us with the *interactive image*. While the other pictures either remain static or change only in a fixed, predefined manner (like film), computers allow us to present pictures that can change almost instantaneously on accord of the beholder following rules determined in advance: Indeed, the “beholders” have now become “users” influencing more or less directly the pictures generated and modified “on the fly” for them – a process depending mainly on the concrete course of previous interaction between a user and the picture-generating system. Quasi instantaneous creation and modification of perspective or illumination, “camera parameters” or elements of the scene depicted, attributes of the depiction styles or even the characteristics of visualization methods of non-visual data can be employed for a wide variety of tasks: illustrating experiments in physics, simulating flights for training purposes, entertaining with computer games, informing by user-adaptive info-graphics, and even providing new artistic experiences in computer art – to name but a few.

In fact, the defining characteristic of interactive pictures had to guide the attention in computational visualistics to the *pragmatics* of pictures. While earlier approaches have mainly concentrated on syntactic aspects (as in image processing) or facets of semantics (e.g., computer vision and computer graphics proper), computational visualistics explicitly focuses on picture pragmatics as the most general approach to images and their uses, which indeed encloses semantics and syntax as significant but restricted parts.

Like linguistic pragmatics, pictorial pragmatics is a waste and complicated field, and even more: a field yet widely unexplored. Nevertheless, many computer scientists have investigated aspects of picture pragmatics, particularly from an application perspective and often with a focus on interactive images. An overview of their experiences, observations, and results thus might help to better understand the current position in computational visualistics and image science, and to plan the next research steps to be sighted at.

The *purpose* of the picture presentation is usually an essential criterion for deciding about the properties of the image to be created. For dealing in an algorithmic manner with the purposes of those involved in the presentation act (i.e., mainly: system designer, content provider, and current user), according "purpose structures" have to be modelled in general and updated to each current case, and parts of that structure must be systematically mapped into fitting parts of the image generation or selection processes. A first step has been made by adapting user models and other strategies derived from pragmatics in computational linguistics for picture systems.

This call for papers asks for contributions to picture pragmatics from the particular point of view of computer science and media informatics. It is particularly interested in pragmatic aspects related to interactive pictures. Papers in German, English or French on themes around the following 'crystallization cores' are strongly encouraged:

- ***"Smart Graphics": Applications of Pragmatics for Interactive Images***
- ***Pictorial Pragmatics for Immersive Systems***
- ***User Modelling for Picture Presentations***
- ***Employing Speech Act Theory for Interactive Pictures***
- ***Interactive Graphics in Mixed-Media Presentations***
- ***Information Visualization with Interactive Systems***
- ***Pragmatic-Based Selection Methods for Picture Databases***
- ***Interactive Pictures and Computer Art***

The thematic issue is particularly intended as the attempt to offer a clear and easily understandable summary of the state of the art of research on interactive images and pictorial pragmatics in computational visualistics for picture scientists of the other disciplines. The authors are therefore advised to keep in mind that they write for an interdisciplinary audience.

Submitted papers should not exceed about 50 000 characters. They may rather be richly illustrated. Peer reviewing follows the system established for IMAGE. Before the final publishing, the authors will have the opportunity to access per Internet the other contributions to the special issue, to comment on them, and to co-ordinate their papers with the others.

Please submit – preferably in electronic form, e.g., as an RTF file – your paper till

September 1st, 2009

to the editorial board of IMAGE or the editor of the special issue Jörg Schirra:

Editorial Board of IMAGE
z.H. Prof. Dr. Klaus Sachs-Hombach
Technische Universität Chemnitz
D-09 111 Chemnitz
Germany
Email: info@bildwissenschaft.org

PD Dr. Joerg R.J. Schirra
Brunnenstr. 19
D-66 557 Illingen
Germany
Email: vertigo@directbox.com

[Inhaltsverzeichnis]

Impressum

IMAGE - Zeitschrift für interdisziplinäre Bildforschung wird herausgegeben von Klaus Sachs-Hombach, Jörg R. J. Schirra, Stephan Schwan und Hans Jürgen Wulff.

ISSN: 1614-0885 URL: http://www.image-online.info/

Verlag
Herbert von Halem Verlagsgesellschaft, Lindenstr. 19, 50674 Köln

Mission Statement

The e-journal IMAGE. Journal of Interdisciplinary Image Science is designed to be an internationally appraised multi-lingual journal for interdisciplinary image research. The variety of disciplines covers all areas of research that are concerned with questions of visual representations. This also includes the discourse between accounts of a more reflexive, theoretical nature and those more practical in aim.

IMAGE intends to promote interdisciplinary research and discourse within image science. It aims to encourage the establishment of an institutionalized general image science. In addition, it strives to encourage the creation of innovative structures in research and science.

IMAGE is supported by the Centre of Interdisciplinary Image Research (ZiB) and – for assessing submitted papers – by a large active advisory board. It is published by Klaus Sachs-Hombach, Jörg R. J. Schirra, Stephan Schwan and Hans Jürgen Wulff in cooperation with the Herbert von Halem publishing house.

IMAGE focuses on methodologically reflective papers that aim to develop an interdisciplinary approach to image science. Initially, contributions will appear at irregular intervals, partly collected in thematically organized volumes. Publication will be announced via the mailing list of the Virtual Institute for Image Science (VIB).

We would very much appreciate the submission of suitable manuscripts. Please send the manuscript to the editorial board by e-mail so that it can be assessed immediately.