EUROGRAPHICS 2021
H. Rushmeier and K. Bühler
(Guest Editors)

Volume 40 (2021), Number 2
STAR – State of The Art Report

WT: A Survey of Control Mechanisms for Creative Pattern Generation

Name[1,2] and Name[2]

[1]Place
[2]Place

Abstract

Alternative title: A Survey of Creative Pattern Generation

***Note:** This is currently a summary of the STAR for communication purposes - not the abstract:*

Supporting artists with meaningful digital tools for creative creation is an ongoing research challenge and spans over various disciplines. On an algorithmic level, most solutions focus on adding singular features and control mechanisms. Little attention is paid to novel methods that can complement each other as part of a cohesive pipeline and creative workflow. Towards the goal of enabling a creative workflow, we survey current methods in the scope of creative pattern generation and their control mechanisms. We classify techniques and discuss their potential to support creative creation with newly developed criteria.

The specific goal of two-dimensional and creative pattern generation, offers a complex design challenge. On the one hand, the repetitive nature of a pattern is well represented with algorithmic creation, especially with procedural representations. On the other hand, the design space of possible pattern designs ranges from realistic, over abstracted to artistic. Here, considerations about a creative workflow and the need to go beyond automation rise and meaningful control needs to be given to artists. Typically however, with the power of procedural generation comes difficult controllability. Combining the algorithmic nature of procedural generation with novel forms of controllability has great capabilities and potential to offer truly novel benefits to traditional creation processes.

For analyzing the state of the art and a better understanding of creative creation with digital tools, we dissect a creation process into to the methodologies of how, what, where and when. To relate control mechanisms on algorithmic level to the stages of a creation process, we classify specific control mechanisms into exemplars, parameterization, handling, filling, guiding and placing interactions. For making the domain of creative creation more manageable, the creative means of navigation, transparency, variation and stimulation are defined and linked to the control mechanisms.

With these criteria, we survey the most recent algorithmic representations for creative pattern generation. For this chosen design goal, the analysis bridges between various techniques such as procedural and data-driven approaches. We identify open issues and sketch out promising research directions towards the unification of different building blocks to a coherent pipeline. All in all, we hope to inspire innovation for artist-centered creation processes on a grander scheme.

CCS Concepts
• ***Computing methodologies*** → *Collision detection;* • ***Hardware*** → *Sensors and actuators; PCB design and layout;*

Important: The status of all references is 2018. All references from 2018 on are still missing!

1. Introduction

Digital tools for creative processes with various forms of output are indispensable for most artists. Even designs that have a distinct hand-made quality to them, such as Disney's animation Paperman for example, were computer-generated [Wal12], employing novel software solutions that are fully controllable by artists. Furthermore, more than three decades of research in academia have produced various control mechanisms for creation. These are often said to be artist-controllable but are less often proven to be so.

Most research has been executed without direct and continuous collaboration with artists. Moreover, large-scale user studies with suitable participants are usually impractical. Due to these obstacles, there is little common understanding in the research community regarding what artist needs actually are and how mechanisms are validated. Furthermore, research has typically focused on solving one specific aspect while accepting significant trade-offs for other steps in a creation process.

For example, the automatic targeted control of a large design space results in long computation times and non-intuitive configuration requirements (e.g., for [BD04; WZS98]). Equally, flexible creation pipelines that enable both automation and manual controllability often suffer from a restricted design space (e.g., for [SP16]).

It is an important challenge to investigate creation processes from a more artist-centered perspective and to consider all tasks and efforts as a whole, from initial configuration requirements to local edits. As a further challenge, control mechanisms have to be linked to an expressive design space, which is the basis for all meaningful creative creation. In order to analyze and validate novel creation algorithms, artist feedback also has to be evaluated. However, such interdisciplinary studies still suffer at times from undefined language and vague discussions. It is an open challenge to complement user studies with a more precise and determinative terminology that is also commonly applicable and understandable by artists.

The previous statement naturally leads to general questions about artist-centered full controllability in combination with distinctive output spaces and a meaningful and applicable evaluation of digital creation tools. These considerations are based on decades of research in multiple disciplines. However, there is little common ground on what is needed as theoretical basis for an evaluation of creative control mechanisms with various open questions. For example, what are relevant characteristics of a creation process with digital tools? What are specific control mechanisms, ranging from global to local and from automatic to manual, with varying levels of abstraction for their handling? How do these mechanisms relate to the stages of a creation process? What are the requirements for creative creation, and how can these be customized to the context of digital creation tools?

To bring further insights, this survey investigates those problems in the scope of creative two-dimensional visual pattern generation, tackling a restricted but highly challenging and representative creation process and design space.

The underlying regularity of pattern designs is based on a repetitive and balanced distribution of elements, usually following hierarchical structures. These characteristics can be efficiently implemented by procedural approaches [ŠBM*10] because they automatically fill a space based on generative rules. An artist should be freed from such tedious, non-inspiring and repetitive tasks. In order to execute order, computational generation techniques are not only an easement, but they also perform in a potentially more precise and less error-prone way than a human artist. Hence, procedural representations are an ideal basis for creative pattern generation. However, the creative demands of, for example, laying out space-specific designs and of placing highlights must also be considered. Procedural models must be augmented, and different approaches must be unified in order to enable the control and quality of manual creation as well as the efficiency and accuracy of computation.

Much effort has gone into investigating control mechanisms within specific contexts, as procedural representations are notoriously difficult to control [BD04; LVLD10; GD10; BŠMM11; LLD12b; LLD12a]. Botanical and architectural procedural modeling, for example, are popular fields of research. There have been summarizing surveys for procedural noise [LLC*10], landscapes [STBB14] and urban spaces [VAW*09] as well as in the context of games [HMVI13; TYSB11], including even a short summary of models for ornamentation [Whi10]. However, the overall investigation of generating and designing patterns creatively is less prominent. This could be credited to creative pattern generation being an ill-defined domain due to it involving creative-artistic considerations. Nonetheless, its diverse design aspects make pattern generation a rich and compelling topic.

Within this topic, this survey present the state of the art in control mechanisms for the creative generation of visual patterns with the following goals:

- The identification of relevant characteristics of a creative creation process with digital tools. A dissection of a creative process into potentially classifiable characteristics as common ground for a discussion about enabling creativity.
- The classification of control mechanisms as theoretical grounding, ranging from global to local and from automatic to manual, with varying levels of abstraction for their handling and the relation of these mechanisms to the stages of a creation process.
- An interlinking analysis of the state of the art, bridging between procedural and data-driven solutions with joint classification results.
- An investigation of the capabilities of a specific mechanism in a creative process and their potential for creative creation within a coherent pipeline.

2. Terminology

Question: Integrate these definitions into their respective sections?

Note: This is probably too long?

The following clarifies the usage of terms that are relevant for an analysis of control mechanisms. Some aspects are discussed in detail in different sections of this report but are included here for an overview of terms.

Pattern: Constitutes a generic term for any type of repeated, often regular, arrangement [Oxf17].

Texture: In the context of computer graphics, *texturing* is commonly understood as modeling a surface's color (i.e., their color texture) with no implications for a design, while designing a surface' s interaction with light is understood as *shading*.

Texture refers in its traditional meaning to the character of a woven fabric [Oxf17] with properties such as *fine* or *coarse*. This work understands texture similarly with regard to potentially repetitive structures. LIN et al. [LHW*06] define a spectrum for such

structures. The spectrum ranges from regular deterministic textures with distinguishable texture elements recurrently placed to irregular placements to purely stochastic textures.

Decor: Refers to elements that generally embellish and beautify without implying any specific design rules in itself.

Ornament: Constitutes a specific type of decor adhering to certain design rules, such as order, hierarchal structures, space adaptation and visual contrast and accents (**?? ??**).

Artistic: Refers to a task with an outcome that potentially has meaning and value beyond aesthetics and practicality. In addition to formal skills that depend on a given domain, an artistic task usually requires creative thinking as well as intuition, emotion and sensual considerations, for example.

Creative: Refers to a task that intentionally produces a novel, non-standard outcome. Please note that this work refers to the academic usage of the term. In common language, a creative task is often misunderstood as one that produces a visual product.

Design space: Refers loosely to all visual results a technique can create. For example, a simple Perlin noise has a rather restricted design space of noise images, only differing, for example, in their frequency. Drawing with a pen can result in many different designs, thus resulting in a larger design space.

Expressiveness: Refers in this work to the size, the variability and the openness of a design space (these terms are in detail discussed in **??**). It is important to note that expressiveness is commonly used in the context of creative controls - however, usually without a clear understanding of its meaning.

Goal-oriented control: Refers to an clearly targeted design task and stands in contrast to exploration. For example, reference images might be given, or an artist may have a clear mindset about how the output of the creation process should look.

User Interface (UI): Refers in this work to a space that is separate from the canvas where an artist controls the system through abstracted representations, such as buttons and sliders or custom-made visual controls. In terms of controllability it makes a difference for an artist to be able to work directly on a canvas or being required to do so in a separated and often abstracted UI and hence this explicit distinction is needed in the context of this work.

Interactive: Refers in this work to systems with which an artist can interact (e.g., through a UI with reasonable response performance). In terms of control mechanisms, we evaluate interactive systems as a whole and also qualitatively. An one-time investment for an initial computation of 10 seconds, for example, is still acceptable, while a 5-second delay at each click is not.

Canvas: Constitutes the area in which the output is generated, similar to a canvas in a painting context.

Shape: Refers to the external boundary or outline on the canvas or of an object without any restrictions on the form.

Curves: Refers in this work as general term for arbitrarily shaped curves or lines without any implications for the formal representation they are based on. Curves can be computed or be derived from drawn strokes, for example. The specifics of these inherently different formal representations are not relevant for the following discussion about control mechanisms.

Procedural: Refers to the production of output by evaluating an algorithm or a rule-based system.

Data-driven: Refers to the production of output based on given, and usually limited, data.

Parameterized: Refers in its original meaning to a system that is based on an implicit equation. However, in regard to control mechanisms and for this work, it simply means that a system offers separated, individually controllable characteristics. Parameterization commonly does not imply a procedural representation but can be part of any technique, including data-driven ones.

3. Taxonomy of Control Mechanisms

A discussion of creative means cannot be derived directly from the related work. Past authors have followed various motivations and have emphasized different aspects when describing their work and results. In order to classify the work in an objective and unified manner, we analyze the actual presented control mechanisms and relate them to general control paradigms. Based on this analysis, we then can also investigate the creative means. While the classification of the control mechanisms can be directly taken from the authors' descriptions (Section 7 Analysis of the State of the Art), its following discussion of the creative means has an interpretative nature to it.

We now lay the groundwork for our later evaluation of control mechanisms of the state of the art. As guide for our review of references, we first dissect a creation process into overall control paradigms and classify specific control mechanisms by their interaction types. The following taxonomy shows the capabilities of the different control mechanisms and potential trade-offs between approaches.

3.1. Control Paradigms

A creation process can be described by answering the questions of *how*, *what*, *where*, *when* and *who*. These paradigms can be discussed in various creation contexts and could even be translated to traditional media such as aquarell on paper.

3.1.1. How

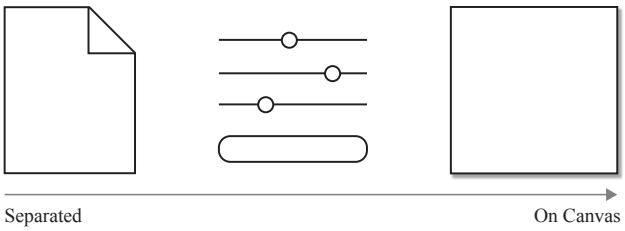
How is a control executed or an input given by an artist? How far is it from the visual result on the canvas?

File: The control is externally given, such as with code or a configuration file.

UI: A separate UI is given through which an artist gives input and activates states. User interfaces are often in close proximity to the canvas, carefully designed and easily usable. However, because they detach the work from the actual output, UIs still have an abstract nature. An artist must actively translate his or her interaction with the UI to the resulting output on the canvas.

On canvas: Controls are executed directly on the output canvas. Most of these controls require an activation or selection of a tool

in a separate UI, such as selecting a pen for drawing on a canvas. In this case we consider the pen primarily as a control mechanism. There are cases where controls cannot clearly be classified as either UI or on canvas. A pen, for example, can have different characteristics that an artist needs to set in the UI. Ideally, the adjustment of settings should be as seamlessly integrated into an on-canvas tool as possible (e.g., with selection choices appearing as tool tips).



3.1.2. What

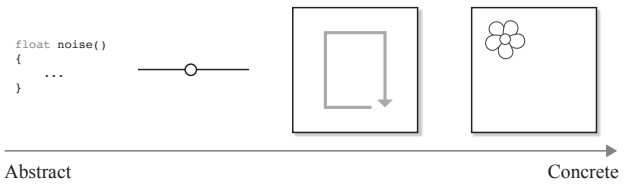
What does an artist give as input? What is the level of abstraction of the content that an artist works with?

Code: Input is a syntactically structured and formal language.

Value: The input is a single value, chosen from a range - for example, with a slider.

Intermediate: The input is visual but still of an abstract nature, such as controlling sketches for a mask or arrows for directionality. Again, artists have to interpret how these inputs affect the result.

Element: The input constitutes a component of the resulting pattern.



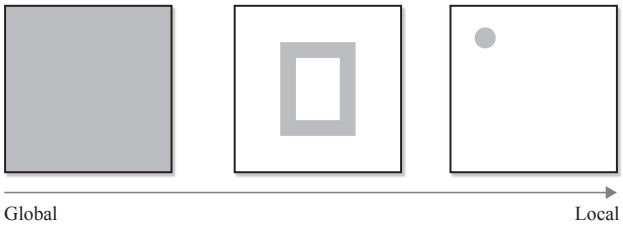
3.1.3. Where

Where does the input have an effect spatially and what is its area of influence?

Global: The input has global influence (e.g., by filling the whole space or by adjusting all elements on the canvas).

Region: The input has an effect in a region of the canvas (e.g., on a drawn curve).

Local: The input has an effect on one specific element.



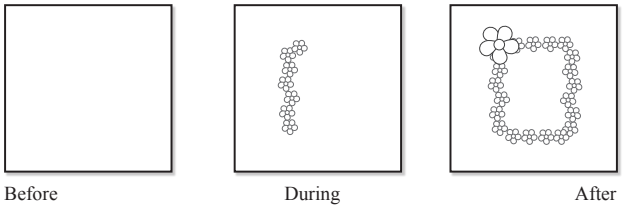
3.1.4. When

When can input be given and at what time in the creation process is the control executed?

Before: Input is given before the actual creation process.

During: Input is given during the creation process, when parts of the results are already visible. This is typically a painting mechanism. However, some processes can also be paused and adjusted.

After: Input is given after the creation process. The result is visible to the artist and can be adjusted retrospectively.



3.1.5. Who

Who can give the input in regard to the type of skill set needed? This category can be in part derived from the above characteristics of *how* and *what*. In most general terms this category can be classified as the following.

Programmer: To give input with dissecting analytical-formal and logical thinking and the ability to abstract.

Artist: To give input with comprehensive intuitive-visual and spatial thinking and the ability to create (e.g., by drawing).

The who category is listed here for completeness. However, to fully answer the questions of needed competencies, skill- and mindsets, including the accompanying psychological and artistic aspects, is out of the scope of this thesis and requires knowledge in fields other than computer science. The following discussions are rooted in computer graphics research and aim for an assessment of algorithmic controllability. Hence, the classification of who specifically is most suitable to use a tool, is put aside.

3.2. Control Mechanisms

For a meaningful analysis, the above classification must be further broken down into the specific control mechanisms.

The following low-level characteristics categorize input modes and their primary effect. Because this survey focuses on interfacing algorithms, UI specifics, such as the layout of buttons, are not considered. Once the specific control mechanisms are analyzed, they constitute a method in combination with the above control paradigms.

3.2.1. Initialization

System Configuration: Required overall setup of the system, such as computing caches or training a model. This is usually a one-time investment.

Task Initialization: A non-creative task that has to be executed each time in order to produce an output, such as selecting the specific optimization algorithm.

3.2.2. Exemplars

Image: An example image that should be matched in its entirety. Examples are usually pixel data.

Element Arrangement: An example element arrangement that should be matched in its entirety. Elements are usually separate shapes and might carry additional data.

Element: One specific asset that becomes in the result part of a whole. Elements can be shapes or pixel data.

3.2.3. Parameterization

Visual Output: Parameters that can adjust visual features directly in the output.

System/Generation: Parameters that influence the output indirectly, such as parameters for an optimization algorithm or constraints.

3.2.4. Handling

Visualization: Any type of visual interface that goes beyond the standard UI elements, such as sliders and buttons.

Image-Based: Images as indirect control input, such as pixel data masks.

Sketch-Based: Sketches and curves directly put on the canvas - for example, the drawing of a mask with a pen tool.

3.2.5. Filling

Shapes: A space to fill (e.g., a specific shape).

Masking: Areas within the shape to fill that should remain unaffected.

Curves to Fill: A one-dimensional curve or path to be filled. The curve is given as a whole before the filling starts.

			HOW			WHAT				WHERE			WHEN		
		Total (out of 40)	File	UI	Canvas	Code	Value	Intermediate	Element	Global	Region	Local	Before	During	After
	Setup														
	Configuration	9	9			8	1			9	1		9		
	Initalization	14	12	2		3	1	5	1	13	2		14		
⟵ *Decreasing Automation*	*Exemplars*														
	Image	1	9	1				1	1	1	2		1		
	Arrangement	9	8	1	1	4		3	6	9	2		9		
	Element	9	6	2	1			5	4	7	2		9		
	Parameterization														
	Visual Output	31	23	8		5	25	2		29	2		17	1	16
	System	16	1	7			15	1		17	4		16		
	Handling														
	Visual UI	6		3	3			5	2	4	5		3	1	4
	Image	9	9			2		7		9	5		9		
	Sketch	7		1	6			8	3	3	7	2	6		2
	Filling														
	Shapes	32	25		8	21		11		32	12		32	1	
	Masking	9	4		6	2		7		6	9		9	1	
	Curve	8			8			6	2	1	8		7	1	
	Guiding														
	Painting	6			6			1	5		6			6	
	Directions	7	1		6			7		7	7		7		
	Placing														
	Element	7			7				7			7	7		
	Drag&Drop	5			5				5	1	4	4		4	1

Table 1: *Prevalence of control mechanisms in the literature: In total, 40 publications are included (the discussed state of the art work). Please note, that the totals of each step (how, what, where, when) can exceed the total of that category as it can be implemented within multiple usage scenarios.*

3.2.6. Guiding

Painting/Strokes to Follow: A curve, usually created by mouse movements or with a stylus pen, that is filled with output elements while the curve is generated - often understood as brushing.

Directions: Visual elements such as intermediate curves, arrows or output components that define directions for the design to follow (e.g., with an underlying vector field).

3.2.7. Placing

Element Placement: The direct placement of components on the canvas as part of the final result.

Element Drag & Drop: Drag and drop of components on the canvas within the existing result.

3.2.8. Control Stages of the Mechanisms

For interrelating the control mechanisms to the control paradigms, we considered the publications that are investigated in Section 7 Analysis of the State of the Art. Due to the diversity of the

underlying methods and the different design goals of the considered body of work, we believe this to be a representative summarization.

Table 1 shows that global, hence automatic, control is usually enabled through intermediate representations, such as an example image, while on the other end of the spectrum, the placement of elements as part of the actual output is local, and automation is lost.

Parameterization and the different types of handling also require abstracted input from an artist, such as the use of a slider. Sketch-based controls, such as an eraser, move the interaction onto the canvas and can make small-scale adjustments. The definition of a space or a curve to fill and masking areas is also usually done directly on the canvas but only influence the output indirectly.

A painting mechanism simultaneously creates the output directly on the canvas but can only do so in a limited region depending on the brush size. All other inputs are typically given before or after the generation of the output.

This classification underlines that a focus on one control type, as is usual in computer graphics research, leads to the common trade-off between global automation and local manual manufacturing. In order to support creative work, control mechanisms need to be combined in a novel and unified manner.

4. Creative Means

Many content generation contexts in computer graphics involve creative considerations. To consider creativity is especially relevant for investigating control mechanisms because the controls provide the means for creative creation. However, creativity is also a notoriously difficult topic to address because it is an ill-defined domain and involves insights from various disciplines.

For creativity related research in computer science, on the one hand, there are efforts to develop algorithms that perform creatively. On the other hand, there is the common goal of supporting human creativity with digital tools, which is the focus of this survey. The goal of enabling human creativity is, for example, well researched in regard to interface design. This is an established field in the human-computer interaction community and based largely on the pioneering work of SHNEIDERMAN [Shn07] about *Creativity Support Tools*. Interface design aspects are included but are not focus of this survey. However, the need for bridging between developing the core algorithms and the interface in order to support creative-artistic intent has been voiced by past research [DHF*17; Ise16; Sal02], and this work wishes to contribute to this effort.

For this survey, rooted in the field of computer graphics, we review potentially relevant characteristics of algorithms that might help to enable artists to be creative. Hence, our focus is on underlying algorithms and their control mechanisms.

The computer graphics community usually addresses questions that solve specific and singular tasks, such as example-based texturing, brush-based modeling or optimizing the structural integrity of a pattern. Such tasks concentrate on efficiency, consist of exact descriptions and are well qualified for algorithmic solutions. Creativity, however, as described in more detail in the following section, feeds on flexibility, diversity, exploration and engagement. In order to support these characteristics, solutions should to be more multi-faceted and creation processes should be investigated as a whole.

As evaluation technique, CHERRY et al. [CL14] presented the quantifiable *Creativity Support Index* (CSI), which has found its way into the graphics community [SLD17]. The index measures how well a tool enables creativity based on a psychometric survey. The development and validation of the measurement dimensions - namely, *exploration*, *expressiveness*, *immersion*, *enjoyment*, *results worth effort*, and *collaboration* - are mainly based on user tests. CHERRY et al. [CL14] quantified the specific phrases participants used to describe a creative process. However, a clear definition of terms like *exploration* and *expressiveness* is missing or the meaning of a statement such as "I was able to be very creative, [...]" is left open.

Also, quantified user study is often not feasible, for example for this survey and for assessing the state of the art. Furthermore, doing so would not be meaningful because the support of creativity is not a goal for most methods. However, most methods do offer carefully developed control mechanisms. We propose it to build a discussion of the means for creativity on the presented control mechanisms in a publication and on the specifics given from the authors. Based on the given information, we reflect on the potential for creative means in a meaningful way, even if creative control was not necessarily the authors' intention.

We derive a working basis of such creative means in regard control mechanisms for our discussion in the following review of relevant references. Our classification is meant as a step toward understanding the creative control options within the current state of the art. In terms of measurement dimensions, it can be seen as an evaluation on algorithmic level and a subset of the more general and user-study-based classification with the *Creativity Support Index*.

4.1. Discussion Basis

Academic discussions about what constitutes creativity have a long history in the field of psychology [Wei06], cognitive science [Bod04] and philosophy [Gau10] and is ongoing.

WEISBERG [Wei06] made the argument that a creative person "intentionally produces a novel product" (p.70). Weisberg explicitly decouples a possible generally accepted value of a product from being the result of a creative process (please refer to [Wei06], p.63, for a detailed argumentation). We follow Weisberg's exclusion of the potential and often diffuse value of the result of a creative process and also focus on the creative intent.

BODEN [Bod10] described *novelty* as a surprising product, one that the creator did not directly anticipated (p.30). BODEN also differentiated between a product being surprising or novel to oneself in contrast to something being universally novel ([Bod10] p.30). In this survey of creativity regarding control mechanisms, we only include novelty in reference to the expectations of a single artist.

The integration of *intention* in describing a creative process is crucial for the development of meaningful algorithms. Weisberg explains that a painter who accidentally stains a painting - a stain which is later applauded by the art world as an innovative technique - cannot be considered a creative result. Hence, algorithms need to

enable artists to follow their intentions with transparent and controllable mechanisms. The idea, for example, of an algorithm producing a large number of random design choices for an artist to choose from contradicts this principle of intention.

Weisberg argues that a creator needs domain-specific knowledge and expertise in order to come up with something novel or surprising. He rejects the common perception of creativity as being an "unfathomable leap of insight" and advocates its systematic accessibility. He argues that the perception of creativity as unfathomable results from missing context and domain knowledge and from neglecting to include the whole process that leads to a novel product. Weisberg's argument applied in the context of control mechanisms leads to requiring techniques to enable artists to fully understand the domain they work with. Cause and effect of interactions as well as the overall options to control the output must be transparent and navigable.

Weisenberg concludes his considerations about creative processes by stating that "you must also work to broaden and deepen your database" [MW09]. Hence, control mechanisms not only need to be transparent and fully steerable for an artist, but they also must offer a large space for an creator to explore. BODEN [Bod10] describes this as a landscape to navigate through. This increase of possible options is a core aspect of many common creativity techniques, such as brainstorming, and must also be used for the development of digital tools [TMNY04]. However a design space not just needs to be big but meaningful and well framed for the domain it represents. It has to provide space to delve into without the danger of getting lost. Classical brainstorming, for example, is on the one hand based on the idea of coming up with as many answers to a question as possible - with no restrictions. On the other hand, a brainstorming session starts with a carefully crafted problem statement, which is supposed to be as precise and descriptive as possible. Hence, human brainstormers intuitively remain in the domain of the problem statement and exclusively offer solutions related to the problem. Therefore in a system that computes options for a design space, all options need to make sense, while "enabling someone to see possibilities they hadn't glimpsed before" [Bod10].

Common creativity techniques often include stimulating constraints [OW10; SVO11; BDH14; Sto05] or motivations to guide the exploration in a specific direction. For example, with the Six Thinking Hats technique [De 85], each hat represents a specific mindset, such as "critical" or "emotional", with which a participant should operate. This technique enables a large variety of possible stimuli and cues such as associations, analogies, abstractions, visualizations and reversals, including purely random inputs. Stimuli are a field of active research and as mentioned above, the usefulness of random cues is doubted. MARKMAN et al. summarize their insights ([MW09], pp.48-69) about cognitive support for creative processes, saying that the pool of random stimuli needs to be restricted to increase the opportunity for novelty and to decrease the probability of misleading failures (p.68).

In a process to enable creativity, the target audience is also an influencing factor. Each skill level requires its own unique type of support. CHERRY et al. [CL14] discuss the handling of different user competencies as example for the importance of balancing simplicity and expressiveness. The authors mention that more expressive but also more complex tools score higher on the CSI. In the following discussions, this survey does not explicitly investigates the appropriateness of a technique for different skill levels (unless it specifically distinguishes a related work, as, for example, in [BWCS14]) but instead focuses on the general suitability of a technique to create the design goal with a reasonable training curve for an average artist. Also, for this first investigation, we focus on support for a single artist but it is worth mentioning that an enabling of collaborative creative work would be a meaningful aspect to add in the future.

To sum up the above review in the context of control mechanisms, mechanisms should offer variation, the chance of steerable exploration and meaningful stimuli, according to the domain a given mechanism serves. For these characteristics, there is no clear translation into quantifiable metrics, such as timings or error rates, which are standardized measurements for productivity [CL14; Shn07].

For this survey, we understand variation as the size of the design space within the context of the technique. For the exploration of different designs we distinguish between the general controllability necessary for navigating a design space ("there are many different roads in the landscape"), and the transparency of that navigation and the understanding of cause and effect when using the tool ("I have the map to the landscape and know how to get from one point to another"). Lastly we reflect on the stimuli of a method and its suggestive capabilities.

The following further specifies the above classification categories - namely *navigation*, *transparency*, *variation* and *stimulation* in the context of control mechanisms. However, at this point, these categories can be seen as somewhat loose and experimental, aiming toward a better understanding of requirements for creative controls.

4.1.1. Navigation

The means of navigation describe whether a creation processes is efficiently manageable as well the extent of the controllability.

- *Interactive:* Refers to a system with ideally no noticeable delays when executing controls and computing results. Lengthy, non-creative configuration requirements are also potentially distracting. Hence, a thorough analysis should consider the whole process an artist has to go through to produce a result.
- *Quantity of Controls:* This category indicates how flexible and controllable a technique is, e.g. by counting the number of different controls that can be adjusted for one output.
  Ideally, this category would refer to the ratio of visual features of the possible output that are relevant to humans to controllable features. This would ensure that the controls cover all necessary features and that they complement each other. However, the identification of generally describable, perceptually relevant visual features is out of the scope of this survey and left to future work.
- *Navigation History:* Describes the ability to go back and forth in one's own creation process, such as using an eraser.

4.1.2. Transparency

The means of transparency describe how clear the understanding of cause and effect within the system are.

- *Control Domain:* Refers to how well controls are mapped to visual features and how well they cover the possible design range of each feature. A high-quality control should not have any overlapping effects with other controls.
- *Control Communication:* This category describes how well controls (e.g., with a visualization and/or little abstraction) represent their effects on the result. For artist-centered tools this could mean that controls should be visual and directly on the canvas.

4.1.3. Variation

The means of variation indicate how visually different the results can be.

- *Size of the Design Space:* A design space is limited if all results look rather similar to each other and are part of a specific design class. A large design space of one technique allows, for example, for different texture classes such as combining stochastic and structural creation.
- *Openness of the Design Space:* Refers to the limitlessness of possible designs and that there is no attachment of the technique to a specific design class. An open design space enables an artist to come up with a distinctive individual style, for example. Different artists can create inherently different and unique results with the same tool if it has a open desgin space.

We do understand that a clear definition of the available different design classes is needed. However, these soleley dependent on the design context.

4.1.4. Stimulation

The means of stimulation indicate how well an artist can enter a pleasurable and stimulating workflow.

- *Immersion:* How natural and enjoyable the usage of a system feels.
  An immersive technique needs to be fluent to navigate, controls have to be intuitive and the design space large enough to not to hit its boundaries while using the tool.
- *Stimuli:* The support to find surprising results - for example, with design suggestions or variations of the input. Options to support stimulation are still underrepresented but on the rise with machine learning techniques. A clear definition of this category is not feasible at this point.

4.2. Summary

In summary, for handling the ill-defined topic of creativity, we follow the definition of creativity as intentionally producing a novel and surprising product. We derive means for creative control from recent research results and relate them to control mechanisms. By this we hope to further a more objective judging of the ability of a technique to support creativity and a detailed comparison of methods and give an first example for that in the following review of the state of the art.

Various aspects of our discussion guides still leave room for interpretation. Knowledge from other disciplines, for example in regard to the perception of visual features, can contribute with valuable insights. We hope that our work inspires such research towards a quantifiable analysis of creative control.

5. Designs

In the pervious section we have categorized control mechanisms and have discussed factors that might enable a creative workflow. In the following section we highlight to which type of pattern designs creative creation could lead.

5.1. Design Goals

The universe is made of repeating structures that can be found on all scales, from the nature of galaxies down to molecular micropatterns. Equally ubiquitous are repetitive structures visible to the human eye. Such patterns range from natural appearances, like stone or wood textures, to highly stylized and abstracted designs, such as ornaments. Artists throughout all cultures and times have used creative patterning and ornaments to embellish the world around them.

On the one hand, creative pattern generation includes repetitive and ordered structures that are often considered as *textures*, thus demanding automatic and procedural creation. On the other hand, creative pattern generation might also include a global layout, adapting to the space they are filling. Furthermore, this type of patterns might include visual hierarchies and highlights that are singularly placed with creative intent. With that creative pattern generation is an interesting testing ground for addressing the delicate balance between giving artists as much control as is needed without burdening them with unwanted details.

Ornamentation is one specific type of creative pattern generation. Even though this survey investigates a more general design space, in the following we are briefly summarizing the specific design goal of ornamentation for a better understanding overall. Ornamentation brings many design aspects of creative pattern generation together and triggers challenging questions.

5.1.1. Ornamentation

Question: Should the following description of ornaments be shortened?

The Oxford English Dictionary [Oxf17] defines ornaments as nonessential accessories intended to adorn. There is no functionality to an ornament other than to beautify a manufactured article without changing its shape or character [War96]. The term ornament can be found in a large variety of contexts, such as in architecture, music or poetry, but this work only refers to two-dimensional visual ornaments. While ornaments may carry symbolic meanings in the arranged elements [Wor96], this work does not include semantics but focuses on visual qualities.

Different cultures and times resulted in various ornamental styles, with great differences in the details as Figure 1 shows. Nevertheless common underlying design principles for ornamentation can be identified.

Ornamentation can be understood as an accurately defined type of decor that follows a structural logic [War96; MOT99; ABS06]. In addition to its aesthetic appeal, an ornament is perceptually distinguished by a sense of order and by its alignment to the space it fills (as summarized by WONG et al. [WZS98] and originally stated by WARD [War96; Dre75; ABS06]).

Figure 1: *Historic ornamentation examples, representing creative pattern generation. Places of origin from left to right, top to bottom: France, China, USA, UK, Egypt, UK*, Poland, UK*, Greece, Italy. *Images are cutouts of tiled pattern but are often found as presented here. Image sources: please refer to* **?? ??**.

An underlying perception of order in an ornament is established by even repetition and a balanced distribution of elements, with an intentionally designed and artificial quality [War96]. Balance can be achieved with a careful composition of elements, and such balance is built on symmetrical arrangements in most ornaments. Compositions are not limited to the repetition of the same element, but different visual qualities can create various relationships. Visual characteristics attract the eye differently and the *visual weight* of a feature can be used as a measure for the degree of attraction. For example, large, dark and highly saturated colored elements have a greater visual weight than small, light and desaturated ones. These visual weights can be used to create visual correlations (for example, based on Gestalt psychology - a topic too wide for a discussion here) and can counterbalance each other. A larger and lighter colored element might have the same visual weight as a smaller, darker colored one. Hence, varied elements with different visual properties can be combined and still make a balanced whole.

Hierarchical compositions further increase a sense of order but are also used for creating contrasts (e.g., foreground vs. background) and accentuating structures (e.g., framing). These structures are often used to elaborate and accentuate the form of the space they fill, building an ornament-object relationship [ABS06]. The following differentiation of an ornamental decoration gives an intuitive understanding of this aspect [ABS06]: Wallpaper can be trimmed for different rooms, but the design is not reproportioned or altered. An ornament, however, is fitted to and references the logic of the space it is designed for. Without adjustment, it cannot be transferred to a different space.

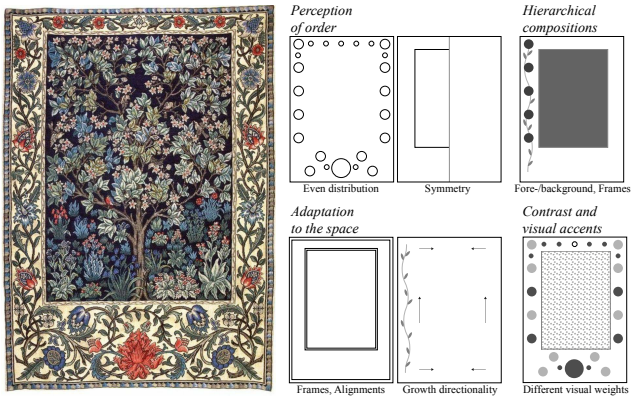


Figure 2: *Exemplary dissection of visual characteristics fulfilling ornamental principles. Single features often support several principles, as, for example, the frames and borders create a hierarchical composition, an adaption to the space the ornament fills and visual contrasts. Image source:* ***[morris_1910_tol]****.*

Contrasts and accents are crucial for the visual appeal of an ornament [WZS98; War96; MOT99]. Single, visually dominant elements and structures might not follow the underlying order of the ornament at all, breaking an otherwise too homogeneous appearance - again distinguishing ornamentation from wallpaper.

Figure 2 gives an example of how the described design principles are combine seamlessly into a coherent design.

It takes artistic expertise to balance the contrast between carefully chosen visual accents and to create a sense of order by applying compositional rules and by complementing the space. However, it is exactly this combination of qualities - rule-based composition and repetition on the one hand and the placement of visual accents and the breaking free from order on the other - that make creative pattern generation an interesting but highly challenging field of algorithmic research in the context of computer graphics.

5.2. Summary

Creative pattern generation exemplifies the common challenge of enabling control for tasks for which humans are indispensable in combination with the automation of tedious manufacturing and the computation of structuring rules. With that it is a representative design goal to aspire to for an general investigation of creative control mechanisms.

6. Models

In the context of computer graphics, generation techniques are usually differentiated into procedural and data-driven approaches. This understanding applies equally to the generation of geometry, animations and texture, for example. Procedural techniques describe the visual output by evaluating an algorithm, while data-driven approaches rely on existing data, such as photographs. However,

through the continuous development of both fields, approaches started to blend and the advantages of both approaches are brought together.

For this goal, this survey focuses on procedural models as a basis, but it also integrates and highlights promising or desirable characteristics of suitable data-driven techniques.

6.1. Procedural

EBERT et al. [EMP*02] describe procedural techniques as algorithms and mathematical functions that synthesize a model or an effect. Solely equation-based representations are considered the "purest" form of procedural modeling [STBB14]. This approach gained immediate importance in the early days of computer graphics. Simple equations are able to reproduce many natural phenomena - such as wood, stone, water, smoke and plants - with only some lines of code in the range of kilobytes, hence being memory efficient. The main appeals of such procedural representation include its compactness in combination with being continuous, scalable and unbound to a specific resolution. While procedural generation techniques have been a constant basis for generating content for games, its characteristics of memory efficiency and unlimited resolution are more important than ever with the rise of virtual reality, for example.

The compactness and efficiency of a procedural model also enable parameterization, resulting in the model being responsive and flexible. Parameters usually represent certain visual characteristics and their amplification. Parameterization brings the crucial benefit that, for example, textures remain editable throughout the entire visual effect production pipeline.

However, the effectiveness of traditional parameterization in helping an artist fulfill design goals is debatable. EBERT et al. [EMP*02] argue that parameterization brings the benefit of a few parameters controlling large amounts of details. At the same time, this is potentially problematic for the realization of specific designs because these often require full individual control of all visual elements. Additionally, parameters are often non-intuitive due to representing overly abstract characteristics of the underlying functions and having overlapping effects [BD04; LVLD10; GD10; BŠMM11; LLD12b; LLD12a].

In addition to the disadvantage in the control of a procedural representation, the creation of the representation itself, the procedural model, requires considerable effort - even though it is only a one-time investment. For the appearance of a model, the focus usually lies on a more generic design, like a texture class. For procedural textures specifically, handling antialiasing efficiently can also be challenging. For a valuable and in-detail survey of function-based design principles of procedural models with focus on textures, the interested reader is referred to EBERT et al. [EMP*02]. Procedural models are not limited to purely function-based designs. For example, the pioneering work of PRUSINKIEWICZ [Pru90] applies the grammar-based L-system to algorithmically model plant growth, an approach extensively investigated by the computer graphics community and considered as procedural.

The classifications of core mechanisms for procedural generation of HENDRIKX et al. [HMVI13] in the context of games and the categorization of SMELIK et al. [STBB14] for virtual worlds are equally appropriate in the context of creative pattern generation. In the following we briefly discuss core mechanisms in the context of creative pattern generation, based on the previously mentioned taxonomies ([HMVI13], [STBB14]).

6.1.1. Stochastic Models

For stochastic models, noise functions generate maps of random values. They can either be used in their original form as procedural model or as a basis function. Visual features can be added by combining multiple layers of the noise in different resolutions.

Typical noise functions are lattice value noise, lattice gradient noise (e.g., Perlin noise [Per85]), sparse convolution noise and spectral noise [EMP*02; LLC*10]. In the context of creative pattern generation, stochastic models build a basis for many designs but their design spectrum and controllability are limited.

6.1.2. Function- and Rule-based Models

Function-based models extend the class of stochastic models by layering and combining a variety of functions to form a visually complex pattern. Typical building blocks are periodic, spline, step, clamp and conditional functions [EMP*02].

Rule-based models are part of individual, and often quite complex, generation systems that can be context-dependent and/or design-specific. Rule-based models are programs that relate to and partition the space to fill and follow propagation rules. The algorithmic core often handles proxy shapes, while for the result graphical elements, such as vector graphics, are mapped to the proxies.

Rule-based procedural models are suitable for creative pattern generation and novel control mechanisms because their iterative generation logic is open and flexible [WZS98; MM12]. They can implement any designs and include any elements. Moreover, within a suitable pipeline, they can potentially take global constraints into consideration and build structural hierarchies.

6.1.3. Grammar-based Models

These models form grammatically-correct sentences from individual words, based on a system of rules. Prominent techniques are L-systems and shape grammars. An emerging subgroup of grammar-based models includes *probabilistic* inference into the derivation of correct sentences from a grammar.

In recent years, there have been a variety of successful grammar-based approaches for certain aspects of creative pattern generation [BŠMM11; TLL*11; RMGH15]. However, grammars are difficult to set up and to design [ŠBM*10]. Because the execution process is inherently hierarchical, grammar systems have difficulty in supporting creative control from a global to local scale.

Simulation Models: Simulation models are based on techniques that approximate complex phenomena for which an analytical solution is unmanageable or unavailable. HENDRIKX et al. [HMVI13] further group simulation techniques into cellular automata, vector and tensor fields, and agent-based simulations.

In the context of creative pattern generation, simulation models are not prominent, with the exception of vector and tensor

fields [IMIM08; LBZ*11; SKAM17]. For simulation models usually interactive performance is a challenge, as well as the control on an element level. However, the potential to create a layout within a space and adapt to the characteristics of that space within a simulation system as well as the possible design variability call for further investigation.

6.1.4. Artificial Intelligence Models

Artificial intelligence models represent approaches that go beyond the direct execution of specific rules. For example, they automatically optimize results based on fitness or error functions, or they apply planning steps. HENDRIKX et al. [HMVI13] group this class into genetic algorithms, neural networks and constraint satisfaction and planning.

In recent years, machine learning has been introduced into procedural content generation with the same impact as in all other computer science fields [SSG*17]. The potential of machine learning techniques in regard to creative pattern generation seems almost limitless and is further discussed as outlook (**?? ??**).

6.2. Data-Driven Models

In contrast to procedural techniques, data-driven methods can be used in two ways in the context of creative pattern generation. First, they describe the processing of input pixel data, such as a photograph. Second, they refer to the output of a method, which is again pixel data. Data-driven models traditionally do not include underlying design models, as procedural representations do. Consequently, data-driven approaches are flexible in terms of possible designs and can achieve photorealism by processing real photographs.

At the same time, photographs bring the disadvantage of potentially including visual features, such as illumination effects, which are unwanted and difficult to remove. Moreover, further down a production pipeline, pixel data is usually not editable anymore. Working with data such as high-resolution images leads to high memory requirements, and without additional algorithms, data is fixed to its given resolution and scale.

Addressing the issue of resolution example-based synthesis is a well established field of research and aims to create infinite amounts of pixel data based on a given exemplar. The pyramid-based texture synthesis of HEEGER et al. [HB95] is an early famous example. WEI et al. [WLKT09] present a comprehensive summary of such example-based texture synthesis techniques, discussing statistical feature matching, neighborhood matching, patch-based and optimization methods. Overall, example-based methods for texture synthesis have achieved similar results in data size, random accessibility and editing and resolution options as procedural textures - but only within specialized contexts and not in an unified manner. Procedural textures offer these capabilities as inherent and combined characteristics.

Data-driven models are numerous and diverse because they can use and produce any input and output data without an underlying procedural model. In the following survey of the state of the art for creative pattern generation, we include various data-driven techniques, however an overall classification is not the focus of this work. Relevant techniques include the tiling and distribution of elements and drawing and brush mechanisms.

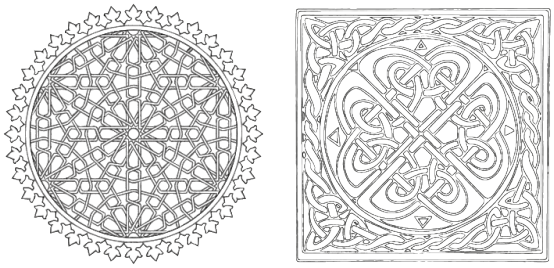


Figure 3: *Examples of traditional Islamic (left) and Celtic ornamental designs. Image sources:* ***[islamic_ornament; celtic_ornament]****.*

6.3. Specific Pattern Designs

In this section we review models that output specific pattern designs, which have as design goal ornamentation, which is exemplary for creative pattern generation. The references in this section summarize work that solely focus on the output and hardly offer any control mechanisms or support a creative creation process. Work that produces similar aesthetics but also offer control mechanisms (e.g., [WZS98; CSC12; ZCT16]), are reviewed in detail in the Section 7 Analysis of the State of the Art.

Work on the generation of pattern designs is vastly spread over various research communities in an isolated and sparse manner [Whi10]. For the development of models WHITEHEAD [Whi10] differentiates between two motivations in the context of generating models for ornamentation in games. First, the author identifies the goal to reproduce existing patterns such as Islamic and Celtic designs. Such work is mainly found in the communities of mathematics and computer science. Second, WHITEHEAD identifies the goal of generating novel pattern designs, which is held mainly by algorithmic computer artists. Such designs are usually not executed in an academic context and, beyond the presentation of the results, are unfortunately not well documented. Only a few exceptions, such as the work of TAKAYAMA [Tak16] in regard to 3D-printed ornate shapes, stand out, and we do not further investigate computer artists' work.

For what WHITEHEAD [Whi10] calls mathematic/scientific ornamentation in the context of games, the author describes tiling and symmetry as the most relevant constituting rules. In combination with interlacing parts of the pattern while repeating and tiling elements, these principles are able to systematically describe Islamic [Ost98] and Celtic [Cro93] patterns (Figure 3).

The seminal work of KAPLAN et al. [KS04] presents an algorithmic representation of Islamic star patterns, a topic still of appeal [KB18]. Readers further interested in this line of work are referred to the dissertation of KAPLAN [Kap02]. ETEMAD et al. [ESP08] and HAMEKASI et al. [HS12] also focus on Islamic flower patterns. Celtic designs were, for example, also successfully computed by KAPLAN et al. and [DS13].

In addition to Islamic and Celtic designs, a variety of other pattern designs have been algorithmically formalized, such as Gothic window tracery [HF04], M. C. Escher patterns [DLW81; KS04], woodwork [GTS10; GSK12], optical illusions [CYZL14], and general patterns [OZH15; Gda17].

6.4. Summary

In summary, in order to produce designs that include not only repetitive structures but also visual hierarchies and highlights for example, most procedural models in their "pure" form are too limited in their design variability. Data-driven models are more flexible in both their design space and creative control mechanisms. However, data-driven models are also tedious and not suitable for automatically executing certain design rules. The following review of the state of the art gives a detailed comparison of the capabilities for creative pattern generation.

7. Analysis of the State of the Art

The contribution of this survey is twofold. On the one hand, its categorizes the state of the art with regard to control mechanisms, translates this categorization to overall control paradigms and discusses their means for creative control. On the other hand, it selects the work not by its underlying algorithms and creation techniques but by its design goal, namely creative pattern generation. This type of pattern generation includes a variation of representative creation challenges, such as combining ordered fillings and repetitive structures with individual global layouts and highlighting components that might break that underlying order. The focus on the visual output instead of specific underlying algorithms allows for a novel and unifying discussion of techniques and merges a discussion of work that is traditionally studied separately. Thus this focus furthers a common understanding of creative control mechanisms.

The analysis of the control mechanisms is directly taken from the authors' descriptions. However, the following discussion of the creative means has an interpretative nature to it. Despite our best efforts to give a clear reasoning, we also agree that this is not always unambiguously manageable. Thus, this discussion should be viewed as a step toward an objective discussion about terms such as artist-usable and creatively controllable as well as a more realistic usage of these terms.

Note: The following categories for the subsections are up for discussion.

Note: The following texts are not reworked and aligned to the storyline of the STAR yet.

7.1. Texturing Methods

Question: Should the texturing section overall be shortened?

Texturing methods focus on creating a repetitive and homogeneous pattern as automatically as possible. These methods usually provide only a fraction of the design space and controllability needed for creative pattern generation. They are applicable for the subparts with a texture-like quality to it, such as background regions and fillings.

But as the investigation of procedural texturing has been the driving force behind the development of procedural representations in general, it produced manifold approaches and noteworthy control mechanisms. Even though texturing methods are not one-to-one transferrable to ornamentation, their rich research history and solutions must be included when investigating procedural ornamentation and might inspire ornamentation specific controllability.

7.1.1. Example-Based Control

Example-based approaches compute a separate output based on a given example and provide a *goal-oriented control*. The motivation behind using these techniques is mainly to generate a specific and predictable output as efficiently as possible. Example-based and inverse approaches have a long history in the control of procedural representations. They remain a dominant research field and are relevant for any discussion about controlling procedural models. In this context, the control of a model often directly derives from a new model definition, and the focus of the related work is usually the latter. In regard to creative control, example-based approaches detach the design task to a data-driven image generation techniques, such as taking a photograph or designing a sample in an application such as Adobe Photoshop or Illustrator.

Relevant factors for differentiating example-based techniques are the size of the design space, hence their expressiveness, performance and initialization requirements. The following investigation is roughly sorted by increasing expressiveness.

7.1.1.1. Stochastic Textures

Todo: Condense and shorten this section.

For procedural texture generation, stochastic textures have been the foundation of both research investigations and many complex models. Stochastic textures are generated with noise functions, and LAGAE et al. [LLC*10] present the state of the art for work before the year 2010. In terms of the controllability of the textures, the authors identify three main approaches. First, the indirect access to the noise through the control of the power spectrum. Second, the direct access to its appearance through function parameters, and third, example-based techniques.

The first two approaches are based on specific function characteristics and are hardly generalizable for creative pattern generation. In this context, noise functions are seldomly used in their initial state but more often as a basis for pattern design. We do not investigate the specific noise function parameters further but only include the overall example-based control mechanism. In addition to performance and input requirements as common characteristics, the expressiveness of stochastic textures can be split into representations that approximate a Gaussian texture and textures including global structures [GLM17; LLC*10].

For Gaussian-like textures, the input analysis and function parameter derivation methods are specific to the targeted noise functions. The method of LAGAE et al. [LVLD10] matches noise bandwidths for isotropic multi-resolution noise with the performance

described as “rapid”, given by GILET et al. [GDG12a] as a few milliseconds. In addition to the cropped exemplar, no artist input is required. GALERNE et al. [GLLD12] present a bandwidth-quantized Gabor noise matched by estimating the power spectrum of the exemplar through its decomposition into a sparse sum of Gaussians. Their fitting performance is about 2 minutes per texture with no input in addition to the exemplar. The noise can be further adjusted with an interactive visual editor in which the power spectrum of the noise is represented by individually modifiable sets of Gaussians. Layers can be rotated, scaled, translated and cloned. Due to the abstract nature of the visual features of a power spectrum (which is used in the editor) and the for artists not directly intuitive connection between a power spectrum and the visual features of the noise, the editor has a strong explorative nature to it. However, as the editing itself is interactive and visually appealing, it is inviting to do so. The same authors [GLM17] introduced an efficient sparse convolution noise based on textons. A texton, which is a bilinearly interpolated function, summarizes the power spectrum of a given texture exemplar, and the final noise is generated by placing and summing up textons. The example match takes a couple of seconds, and no further artist input is required.

By introducing a noise that permits for the approximation of arbitrary spectral energy distributions, GILET et al. [GDG12a] increase the expressiveness of their model toward more structural texture designs. For a straightforward noise by example computation, the method of GILET et al. [GDG12a] successively decomposes noise frequencies to match the power spectrum of a multiple kernels noise, and, depending on the number of artist-defined convolution noises, it takes up to 20 seconds. For greater control and expressiveness, a perturbation function and a multi-layer approach are presented. The perturbation can be an additional artist-defined image map of the desired pattern, showing how the pattern should be repeated, breaking the regularity and possibly creating a more structural pattern. How the magnitude of the perturbation is defined is not described by the authors, but it could easily be an input parameter. Furthermore, GILET et al. [GDG12a] interpret texture design as hierarchical composition and employ a layer function that assigns positions to different textures. For this function a artist-defined map can be used.

Further pursuing the topic of greater expressiveness and a more structured noise, GILET et al. [GSV*14] introduced a local random phase noise. The key aspect of their approach is the separation of structure and noise. The noise function itself blends a sum of cosines with random phase, locally centered on a regular spatial grid. A artist-controlled parameter relates to the number of cosines and the visual quality of the noise. Examples of Gaussian patterns can be given by a spectrum or a discrete noise image. For structured designs, specific phases in the power spectrum are fixed independently from the spatial domain. The amount of structure in comparison to noise is controlled with a parameter by the artist. The authors do not report performance times for the matching step. PAVIE et al. [PGDG16] also focus on extending the expressiveness of noise-based representations. The authors argue for control mechanisms being more intuitive in the spatial domain instead of the commonly used editing of the power spectrum. Local random phase noise [GSV*14] is extended by aligning the noise on a regular grid with a spot noise model based on a random distribution of structured kernels. The artist has interactive control of the spatial structures by modifying the spot functions and their distribution, thus increasing the range of possible designs.

GUINGO et al. [GSDC17] base their work on an underlying novel noise model and a separate handling of structures with a bilayered setup. Their method improves spatial variation and visual quality in comparison to other methods. A spatially varying Gaussian noise can be matched to a suitable exemplar by computing a structure layer, extracted by filtering the exemplar, blending masks derived from clustering the spectral domain and different spectra from an auto-correlation technique. In order to procedurally represent the discrete structure and mask layers, a method based on tiling is applied. In order to control the synthesis, the artist needs to adjust two parameters, the number of different random patterns in the input and the size of the local spectra weighting faithfulness to spatial variability of the exemplar. The performance of matching a 512×512 input image can take up to 1 hour (with the current implementation not parallelized). KANG et al. [KH17] decompose the power spectrum of an input image into so-called "feature" and "non-feature" parts. Non-features are obtained by a noise-by-example method. The authors do not mention whether the noise can be further adjusted. Feature parts, such as edges, can be edited in the feature image and are combined with the noise based on a artist-controlled ratio. For the procedural representation of the feature parts, the authors employ data-driven tiling. The feature extraction for a 257×257 input image, and therefore the texture matching, ranges from few seconds to 2 minutes, depending on an additional frequency clustering exploiting spatial coherences in the input. GILET et al. [GD10] apply a more general optimization strategy for choosing the parameters of a noise-based procedure. They minimize an image distance metric computed with a multi-resolution Gabor filter bank and a windowed Fourier transform with gradient descent. With the help of the artist estimating the light source direction in the input, GILET et al. [GD10] can create displacement map textures, with the parameter computation taking from 1 to 3 hours. With a given rough approximation of the geometry and choosing a representative pattern patch in the input, even volumetric representations can be created from the exemplar.

7.1.1.2. Unrestricted Texture Designs All the above discussed noise-based methods control a single stochastic procedural model. Even though recent advances greatly increase their expressiveness, the design space of noise-based models is too limited for creative pattern generation. In addition to methods dealing with generalizable stochastic models, methods employ procedural textures optimized for specific design goals. These textures can potentially be visually more complex to meet the requirements of their intended task. For brick and wood textures, the early work of LEFEBVRE et al. [LP00] presents an example-based control by transferring specific measured properties of an input to corresponding parameters for the procedural representation. The algorithm takes a suitable reference image, a binary mask, and the texture class as input and produces results for these two structural texture types. The authors describe the matching performance from a few minutes up to an hour.

[GDG12b] focus on the interactive creation of procedural semi-structured texture models. We include their work in this discus-

sion because it also handles the control of visual features. With an improved point distribution function that can consider hierarchical spatial relationships, random variations of statistical shape models are generated from artist input. In order to do so, an artist needs to give multiple exemplary object distributions. BOURQUE et al. [BD04] allow for the whole procedural texture spectrum with their parameter retrieval technique. In so doing, they employ two types of similarity metrics, one based on the Fourier transform of the images and the other one utilizing histograms of the Laplace pyramid of the images. For optimization, they apply the Nelder-Mead and the gradient descent method. As input, an artist needs to individually select the distance metric and optimization strategy for each fitting task. As initialization for the optimization, the authors propose "on the order of 200" pre-computed random choices to choose from. The authors report an average optimization time of 12 minutes, not specifying for how many parameters. GILET et al. [GDG12a] report more than an hour for the performance times. For such a search-based approach, the parameter count is highly influential on the performance for both visual quality and computation time. With a higher number of parameters the current form of the approach quickly becomes unfeasible.

7.1.1.3. Element Arrangements An example-based control can be used to arrange elements, which refer to individual visual entities that are the smallest unit for these techniques. From an example arrangements, relationships between elements are extracted, and results are reproduced for the synthesis. Arrangements are often function-based distributions and hence can be considered rule-based procedural models, while the elements themselves usually come from input data, such as vector files. Because many visually complex pattern contain areas of formal arranged elements, this is a relevant sub-goal for a creative pattern generation task. BARLA et al. [BBT*06] and HURTUT et al. [HLT*09] focus on example-based element arrangements of stroke-based vector elements. BARLA et al. [BBT*06] map vector data to an intermediate representation based on proximity and continuation, which the authors call clusters of strokes. To synthesize a similar arrangement, elements are transferred by local neighborhood matching to a global seed distribution computed by Lloyd relaxation. Computing arrangements takes up to 10 seconds, and artist-input is used in addition to the stroke patterns. A choice between two modes for processing strokes and the amount of variation added is a post-processing step. HURTUT et al. [HLT*09] extend that work by categorizing elements as appearance units and transferring their spatial statistical interactions to new arrangements in the order of seconds, also being able to capture non-uniform distributions. As a possible artist input, one exemplary shape input and density map are shown, and other input options are discussed in principle. The authors clearly state their focus to be on automation.

IJIRI et al. [IMIM08] analyze a given element distribution by local neighborhood comparisons and synthesize output with interactive performance with incremental rule-based local growth. Hence, the technique combines data-driven texture synthesis with procedural generation. Element attributes that go beyond the positions of the elements and orientation cannot be controlled. Artists can choose between three element orientation modes, and as a global design constraint, artists can use an interactive spray tool to define areas to grow in, a flow field tool to define overall alignments and a boundary tool. Moreover, the reconstructed topology can manually be adjusted. The combination of tools that allow the artist to work on the canvas support the immersion in the creative tasks because an artist can think less about abstract setups and instead focus on the actual output.

The technique of MA et al. [MWT11] is based on a sample of a discrete element distribution and an output shape to fill both in two and three dimensions. The exemplar has to contain the actual elements in their domain and cannot be basic pixel data. In its broadest sense, this underlying distribution model can be seen as a procedural model. Even though there are no generative rules, characteristics of the discrete elements and their distribution can be parametrized, and changes can be automatically processed and reproduced in the output. In order to fill the output shape with elements, an energy optimization is processed with a novel neighborhood similarity metric. In addition to element positions, the metric includes variable features referring to orientation, geometry, appearance and type, for example. Hence, the metric is capable of reproducing global aggregate distributions that go beyond local element placements. The authors also extended their work to the spatial-temporal domain [MWLT13]. In regard to the available control mechanisms for artists, necessary inputs are the exemplary element distribution, the neighborhood size to consider and the output shape. Further distribution constraints based on element attributes are optional. Examples for the inclusion of a vector field and element drag and drop are given. The authors report seconds to minutes for performance times with a non-optimized implementation.

7.1.2. Grammar Generation

Grammars are a classical procedural representation. Grammar-based output can be designed through the generating grammar, the included visual elements and often custom-made parameters for visual features. To translate a desired visual output to an abstract grammar is a daunting task for most artists, and first efforts have been made to provide an example-based technique for the grammar generation itself. ŠT'AVA et al. [ŠBM*10] present a context-free L-System that is able to recreate a given two-dimensional vector image consisting of groups of line segments. The algorithm creates similarity groups of these basic elements, computes spatial relationship clusters and iteratively translates these into rules. An artist is required to define a similarity threshold and significance weights for the different clusters, such as element distance or similarity, for example, thus guiding their representation according to the L-system rules. The time needed for the inverse step, depending on the number of elements in the input, is reported to range from a few seconds up to 20 minutes. TALTON et al. [TYK*12] further generalize the idea of inverse grammar generation and interpret it as a probabilistic interference problem. Their system induces a probabilistic formal grammar from a hierarchy of labeled components with a Bayesian model merging technique.

7.1.3. Creative Means Discussion

The investigation of example-based techniques shows valuable achievements for goal-oriented control and for increasing design spaces within specific contexts. With regard to creative control, in

addition to the gain in *variability* being a crucial step, the presented work also improves *navigability* through interactive performances.

Element arrangements potentially enable greater visual variation because they do not need to adhere to any rule formulation. At the same time, the generation of a sample arrangement, usually done in an external application, is potentially tedious. A sample that is too small might lead to uniform results. Moreover, elements are often carefully connected in creative pattern designs, and might include a hierarchy of structures, such as for ornaments. Hence, element arrangements can only provide a subset - albeit an important one - of the design space needed for creative pattern generation.

The related work is overall uniform in working toward the classical requirements of finding the most efficient goal-oriented control shows. With the exception of IJIRI et al. [IMIM08] and GALERNE et al. [GLLD12] little effort has been made towards improving visual control for an artist. MA et al. [MWLT13] present various powerful control functionalities but do not show their capabilities within an artist usable scenario. GILET et al. [GDG12b] also offer comparatively more mechanisms but as required configuration for their computation not necessarily as variable controllability.

Even if they are example-based, many techniques still require considerable non-creative effort for an artist, such as working with a power spectrum or predicting how changes in the exemplar, such as element arrangements, affect the output. The potential of these methods for creative control lies in furthering interactive performance, reducing initialization requirements and experimenting with the spatial influence of controls. The presented work only focuses on global designs, such as the whole canvas and repeating regions. Methods for which regions could be defined, models layered or the placement of single elements integrated constitute valuable directions for creative control mechanisms.

7.2. Shapes and Masks

The most basic control requirement is to define an area to be filled. Methods that focus on the development of novel underlying procedural systems with no acknowledgment of control mechanisms, also need to define the space to fill and a relationship of the system to that space. Many approaches take the idea of simply outlining a space further and carefully design growth constraints, offer masking and the sketching of areas to be filled, thus leading to complex designs.

The following section discusses procedural techniques that consider global shapes and masks. One group of methods transform growth constraints to a probabilistic inference problem. Due to the decorative quality of the results and their parameterized control, a data-driven approach is also included

7.2.1. Rule- and Grammar-Based Methods

WONG et al. [WZS98] introduced a programmable procedural system that employs a greedy rule-based strategy to generate floral ornaments. A procedural model is created with artist-defined elements and with a set of growth rules that handle the selection, appearance and connections of elements. The process iterates, finding tentative places for elements by testing them against constraints in the procedural model and, where suitable, placing elements in the found spaces, optionally and connecting them to existing elements. Possible ornament designs are technically restricted only by this iterative creation logic. All adjustments to the design and layout of an ornament have to be done by writing code, with the exception that a "region specification" for the filling can be given. The authors do not report any performance times.

SANTONI et al. [SP16] present the procedural generation of tangles, which are repetitive black-and-white hand-drawn patterns made from dots, straight lines, simple curves and circles. Tangle elements usually align to the shape they fill, for example, by outlining it. A stochastic group grammar with grouping, geometric and decorative operators composites recursive patterns at different scales, filling two-dimensional shapes as well as handling holes. A tangle generation usually takes a few seconds, with a complex example taking about 3 minutes. The authors demonstrate the applicability of their method with an interactive system based on a parameterized artist interface, including history navigation, rule re-expansion and sketch-based operator modification. A user study evaluates the system as accurate, controllable and easy to use after a reasonable training time.

LOI et al. [LHVT17] present a procedural framework for a large variety of element texture designs. The authors aim for designs that are unrelated to their spatial location and the space they fill, calling it stationary. Their programmable method is developed for technical artists and requires programming expertise. Generating pattern scripts are built with partitioning, mapping and merging operators. These operators enable both global and local design control and the composition of designs. The operator-based technique would enable a node-based interface design, which is not explicitly demonstrated in the article. The execution time for most designs is a few seconds, with some examples taking more than 1 minute. A user study with technical artists carefully evaluates the system's scripting interface, concluding positive results overall.

LOI et al. [LHVT17] offer a complex shape-filling and masking system for procedural open L-system models by dividing a target space into artist editable guide shapes. Seeds for the L-system are interactively given by an artist as a position and orientation. The guide shapes determine what types of patterns grow in different areas. The connections between the shapes are manually specified by the artist and in turn guide the connections between elements. Based on a mass-spring system, the guides can be intuitively edited as a whole. The authors report on pattern generation performance for most scenarios as less than a second, with up to 45 seconds for only one complex scenario.

7.2.1.1. Probabilistic Interference Other systems provide global outlining shape control on procedural processes by interpreting the modeling task as a probabilistic inference problem.

TALTON et al. [TLL*11] present for grammar-based procedural models, as example for their flexible analytic objective functions, non code-based global controls through image and volume matching. The authors stress that in principle any control mechanism can be matched with any grammar through their decoupling of the growth control from the grammar itself. The authors discuss that to come to the desired design goal, some experimentation might

be needed, making the approach less transparent. Performance depends on the complexity of the grammar and the number of optimization steps needed. The authors report performance times ranging from a few seconds to several hours. For their examples, the authors manually terminated the optimization iteration.

RITCHIE et al. [RMGH15] controlled rule-based hierarchical and iterative procedural models similar to TALTON et al. [TLL*11] with image-based matching and target volumes. The authors present a sequential Monte Carlo variant that is able to score incomplete model states, thus improving convergence behavior and final scores. The reported performances range from around 3 seconds to 12 minutes, and the authors show that the number of included primitives scales reasonably.

7.2.2. Data-Driven Fillings

For filigrees, which are thinly structured repetitive patterns, CHEN et al. [CZX*16] present a mainly data-driven approach. Their method automatically distributes and assembles a set of suitable independent input elements for which an up vector is specified into a pattern in both 2D and 3D. The authors implement an optimization of a packing problem under specific constraints, mechanically creating strengthened fillings. This method does not rely on an underlying procedural model, but it also processes control parameters for the filigree generation. Due to the nature of the optimizations, a randomization and a distortion ratio parameter are required input. Additionally a field of directional strokes can be drawn on the canvas, controlling element orientation and size. When multiple elements are combined into one common pattern, percentages for appearances of the elements can be given. The performance in two-dimensional space runs from 6 to 26 seconds.

7.2.3. Creative Means Discussion

The above discussed procedural generation techniques offer novel systems that decouple control mechanisms from the implementation of individual models. This enables more possible results for one specific technique, thus improving the size of design spaces. Sophisticated masks and growth constraints lead to visually interesting and complex designs. However, it is not directly predictable how a space will be filled exactly. Because most of the presented methods only offer quite limited interactive performance, even a basic trail and error exploration is hardly feasible; hence, the navigation of the design space becomes cumbersome, and stimulation becomes hindered. The one technique [SP16] that offers the means for a transparent navigation is also the one with the most restricted design space. SANTONI et al. [SP16]s' consideration of a navigation history stands out from all related work in this survey. In terms of stimuli, the mass-spring system for editing control guides offered by BENEŠ et al. [BŠMM11] is a promising direction because it is intuitive, enjoyable to use and encourages exploration.

In terms of control mechanisms for a more complex design goal, these techniques do not permit hierarchical or element-level local controls or the control of element connections needed by artists who want to generate patterns creatively without having to write code.

7.3. Vector Fields

Fields constitute a powerful tool for combining an automatic procedural filling by individually designing regions on the canvas. In the context of two-dimensional ornamentation, these fields are usually vector fields. The streamlines of a field can create curves as part of the pattern that fill and structure a space. The design of a vector field requires less manual work than the manual creation of curves. Other global design choices, such as an overall growth direction or the alignment of elements, are simple to translate from a vector field to procedural generation rules.

Already included in the described work of IJIRI et al. [IMIM08] above, the authors employ vector fields to define the overall growth direction and alignment of elements within an example-guided arrangement.

LI et al. [LBZ*11] present a shape grammar that is guided by either a vector or tensor field. The field can influence the grammar's translation command, potentially leading to globally pronounced structures. The field can furthermore guide rotation, scaling, and color parameters. The artist can specify a priori field constraints, such as regular and singular field elements, on the surface to be filled. Once the field is computed, local Laplacian smoothing can be applied. The authors report a synthesis performance for geometric surfaces from less than a second up to 3 minutes.

SAPUTRA et al. [SKAM17] optimize a flow-based ornamental packing of elements into a two-dimensional outline. For each element, a predefined spine controls the element's deformation. The artist defines direction guides and optionally fixed elements that control the computation of evenly placed streamlines. Elements are placed and deformed along streamlines. An iterative refinement step optimizes for a dense and balanced filling. First, streamlines are slightly shifted to cohere to the space available. Second, elements a re-placed with rotational adjustments and possible overlaps into free space of neighboring elements, reducing negative space. An average packing takes about an hour.

Vector fields are further employed in various other specific procedural modeling contexts. For example for procedural street modeling [CEW*08], micrography [MBS*11] or botanical models [XM15].

7.3.1. Creative Means Discussion

The discussed work shows that fields allow for greater visual variation by opening the design space and transparent control for filling a space automatically. When designing a vector field, artists do not work with the pattern directly, but fields are intuitive to understand. Their abstraction translates to the model in a straightforward manner. Thus, using flow within a vector field to design is a suitable control mechanism, especially for designs that aims to align their elements to the space.

7.4. Curves, Sketches and Painting

Curves and hand-drawn paths give an artist more direct control than the previously discussed methods to fill a space. In addition to the visual output being further constrained, the control is put onto the

actual canvas. Curves are needed for tasks such as creating an decorative frame or structuring the space. Some techniques consider the whole curve before computing the ornament, optimizing the filling of the curve based on certain design goals, enabling a form of global planning.

Painting-tool-like methods create output along curves but do so directly without taking an a priori completed curve into consideration, as if using a spray can or a brush. Painting techniques usually include a brush diameter, hence the size of the area to be filled along the curve.

Besides the value of the indirect use of curves as a control tool, their direct employment as a visual element is also relevant. Formed curves, such as circles, spirals or hearts, are essential components for many pattern designs such as ornamentation.

The following section first discusses work that enables the direct control of curves as pattern elements. The state of the art using curves as control mechanisms are then discussed for both procedural and data-driven methods.

7.4.1. Curves As Basis Elements

ANDERSON et al. [AW08] adapted the design principles for ornamentation discussed by WONG et al. [WZS98] as well as their core growth mechanism of "finding the largest space to fill next." ANDERSON et al. [AW08]s' technique places discrete elements on the sides of an artist-given curve, while not filling the curve itself. The artist can input masks not to be filled, proxies controlling the size and type of elements to be placed and to equal the sum of radii on both sides of the curve. Two input interfaces exist, the interactive view and the buffer view. The authors do not report a user study or specific performance times but call their system interactive.

Also incorporating an artist-defined curve as the spine of a pattern, CHEN et al. [CSC12] use an interactive L-system to attach decorative spiral designs to the curve given by an artist. XU et al. [XM09] use the space-filling algorithm of WONG et al. [WZS98] in combination with particle tracing in simulated magnetic forces for the generation of decorative curves. The physical properties of the charges, the magnetic field and the initialization of the particles are the parameters for designing the curves. The computation takes less than 5 seconds. The authors acknowledge the non-intuitive parameterization of the system and give an example timing of 2 minutes for finding the parameters of a specific example. MERRELL et al. [MM10] generated a set of curves in the same style of a given parametric example curve. A style is defined by local properties, such as tangents and curvatures that are derived from a local shape analysis. The new curves are computed with a rule-based system that allows artists to interactively edit the result. Interactivity is somewhat diminished by computation times of a few minutes for a curve set. ZEHNDER et al. [ZCT16] provide artists with a tool to directly assemble structurally sound curve networks on a three-dimensional surface. The components of the network are spline curves defined by the artist. Components can be placed manually or are repeated semi-automatically. The curves can be moved on the surface while having an elastic quality to them. To prevent structural weaknesses, the system indicates problematic areas and suggests improvements, seamlessly combining the design task with engineering requirements.

7.4.2. Curves As Control Mechanism

MĚCH et al. [MM12] present examples of painting methods for different aspects of generating procedural models, from painting growth constraints, such as masks, to having a pattern grow along the strokes. This discussion only refers to the actual examples given by the authors. However, these are only selective examples for the flexible *Deco* procedural engine. The engine opens up and generalizes environments for interactive control mechanisms for various types of procedural models. For the programming of decorative pattern models within the engine, helpful functionalities, such as symmetry objects and control guides, are predefined. All artist control mechanisms have an interactive performance. Overall performance mainly depends on the pattern generation scripts. The engine offers to load pattern codes as a dynamic library, optimizing performance. In theory, the Deco engine could allow for the editing both of single elements and their connections. This is crucial for decorative patterns, for example, in setting visual highlights. In MĚCH et al. [MM12] however, no examples for this feature are given.

JACOBS et al. [JBMR18] developed the programming and drawing environment *Dynamic Brushes*, in which an artist can create individual procedural brushes for a stylus pen. General programming logic and relevant mathematical functions for creating patterns are translated into a visual programming interface. The evaluation of the system by two professional artists shows that once initial struggles to learn the system were mastered, the artists were able capture their personal analog styles with the procedural brushes. Overall, the authors and the artists open many valuable questions about the usage of current tools and about alternative approaches that seek to seamlessly blend manual and procedural creation processes.

More painting-like methods can be found, for example, in procedural botanical modeling [APS08; CNX*08; PHL*09], procedural landscape generation [EVC*15], as part of a procedural water color engine [DKMI13] or for dynamic effects [XKG*16].

7.4.2.1. Data-Driven Approaches In order to create an ornament along a sketch, LU et al. [LBW*14] present a data-driven approach. Given vector pattern exemplars are placed and deformed along a artist-given curve. Boundaries between element segments and visual soundness are optimized through graph cut and hierarchical texture synthesis. For the exemplars, an artist has to define the start and end point of their spines. If needed, the whole spine can be sketched as an input. The artist can refine results with add and erase constraints that are drawn on the pattern. The authors report a synthesizing performance from 1 to 8 seconds. A related data-driven approach for synthesizing example-based vector patterns along a curve was presented by ZHOU et al. [ZJL14] in the same year. In this work, the authors focus on ensuring a structurally sound output pattern and an extension to fill a surface. Topology descriptors and artist-given topological constraints are included in the element assembling optimization process. Additionally, local pattern orientations and a variation value can be defined by an artist. Once a pattern is generated, an artist can interactively adjust the underlying curve, with the pattern being updated accordingly. Generation performances are reported to be around a few seconds, with complex models a little more than 2 minutes.

KAZI et al. [KIZD12] present a multifaceted tool to create tex-

tures from pen-and-ink drawings with sketch-based control mechanisms, mixing data-driven and procedural modeling. Basis drawings can be repeated along paths, used for brushes, fill regions, optionally consider perspective and propagate modifications of the drawing to all repeated elements. A user study confirms the system's usefulness to efficiently create repetitive textures while maintaining the natural workflow and artistic control of an artist. XING et al. [XCW14] build upon that work by automatically detecting and suggesting possible repetitions to the artist, aiming for a less regular, more painting-like quality. The presented system also offers various brush options and navigation tools in order to combine automation with artist control.

Similar approaches have also been investigated in the context of texture painting [LFB*13], creating mosaics [Iga10; AGYS14] and data visualization [XHC*18]. These ideas cohere to the needed control principles for creative creation while focusing on their specific design tasks.

7.4.2.2. Feature Exploration Even though not a generating technique in itself, exploration is an important characteristic of a creative process. TODI et al. [TWO16] present a tool for exploring sketches and the automatic optimization of common layout types. With the method of CHEN et al. [CFA16], an artist can browse a collection of texture images by sketching highly abstracted pattern features. The represented structural features of reflection, rotation, and translation symmetries adhere to important design principles for visually pleasing patterns. One could imagine a similar intuitive approach for exploring the parameter space of an ornamental procedural representation, for example.

7.4.3. Creative Means Discussion

Curves and sketch-like methods offer a well communicated, hence transparent navigation. The discussed techniques are mostly interactive, artists are familiar with their functionality from the real world and they work directly on the canvas. The ease and directness of usage also constitute a foundation for possible immersive flow of work. Using painting-like methods can allow for smoother navigation by integrating brush settings and increasing the quantity of controls.

However, because creation techniques and design spaces are open, it could lead to manual and tedious creation requirements for patterns, such as when filling a background. Here, the incorporation of procedural creation principles for automatic fillings into a data-driven process by KAZI et al. [KIZD12] and XING et al. [XCW14] is a promising direction. Instead of fostering a free painting-like quality, design principles for pattern designs could be added to create a more organized output.

7.5. Element Placement

The placement of single elements onto the canvas maximizes artist control and is on its own a trivial data-driven control principle. However, in combination with procedural modeling, this mechanism becomes interesting. Separately placed elements that do not follow any rules should be integrated and processed to remain part of the underlying global scene structure. Even though this functionality can be compared to using the tip of a brush, paint-like procedural modeling techniques often have a more spray-can-like quality [MM12] and do not include this option.

For creative pattern generation, this type of variation is needed for preventing a monotonous, texture-like output. The placement of single elements as highlights should visually break the underlying order of repetition, while still being a homogeneous part of the global layout. The following work presents steps toward this demanding goal by integrating the placement of single elements into a global control mechanism.

By detecting symmetries and curvilinear element arrangements in a given vector pattern, YEH et al. [YM09] extend the manual data-driven design processes with procedural-modeling-like editing options. Based on the detected element groups, an artist can adjust the spacing, location and scale of one element directly and propagate that change to the all other elements in the group. The authors also offer a brush that recreates recognized element groups.

The technique of GUERRERO et al. [GBLM16] offers suitable design variations of the vector pattern an artist is working on. An artist can select and continue with one of the offered alternatives. The system constantly re-selects from an exponential number of relevant variations based on the artist's modifications. The user interface is carefully laid out in order to offer design variations in an intuitive and efficient manner while at the same time not hindering an artist's own workflow. The authors thoroughly evaluate their system quantitatively and qualitatively - for example, with a user study. Overall, participants agreed on the usefulness of technique.

7.5.1. Creative Means Discussion

The discussion of this section is closely related to the data-driven sketch-based techniques, and it shows a further promising approach for integrating procedural modeling functionalities into a data-driven process. GUERRERO et al. [GBLM16] present a overall transparently navigable and stimulating control mechanism. With a carefully designed workflow, it further fosters an artist stimulation by offering novel but suitable design variations.

8. Discussion

[...]

Todo: Discuss Table 2.

8.0.1. Outlook

The review of the state of the art shows that there are various limitations and possibilities for future work within the specific contexts of the work. However, there are also novel paradigms for creative control and their underlying algorithms that uniquely add to the state of the art. The most prominent development is the integration of machine learning techniques, which are discussed in more detail in the following. More insights regarding the possibilities of the usage of semantic attributes are also given below.

In addition to machine learning and semantically driven approaches, collaboration is a valuable future line of investigation for enabling creative work. With regard to technology, more and more aspects of common tools either are fully browser-based or are in

CONTROLS	INIT.		EXEMP.			PARAM.		HAND.			FILL.			GUIDE.		PLACE.	
	Configuration	Initialization	Image	Arrangement	Element	Visual Output	System	Visualization	Image	Sketch	Shapes	Masking	Curve	Painting	Directions	Element	Drag & Drop
EXAMPLE BASED																	
Stochastic																	
LAGAE et al. [LVLD10]			×			×					×						
GALERNE et al. [GLLD12]			×			×		×			×						
GALERNE et al. [GLM17]			×			×					×						
GILET et al. [GDG12a]		×	×			×			×		×						
GILET et al. [GSV*14]			×			×	×		×		×						
PAVIE et al. [PGDG16]					×	×	×		×		×						
GUINGO et al. [GSDC17]			×			×	×		×		×						
KANG et al. [KH17]			×			×			×		×						
GILET et al. [GD10]		×	×			×				×	×						
Unrestricted Designs																	
LEFEBVRE et al. [LP00]		×	×			×			×		×						
GILET et al. [GDG12b]				×	×	×					×						
BOURQUE et al. [BD04]	×	×	×			×					×						
Element Arrangements																	
BARLA et al. [BBT*06]		×		×		×											
HURTUT et al. [HLT*09]				×					×		×						
IJIRI et al. [IMIM08]				×		×	×				×	×		×	×		
MA et al. [MWT11]		×		×		×	×				×				×		×
GRAMMAR GENERATION																	
ŠT'AVA et al. [ŠBM*10]		×		×		×	×				×						
TALTON et al. [TYK*12]		×		×		×	×				×						
SHAPES & MASKS																	
WONG et al. [WZS98]	×					×					×						
SANTONI et al. [SP16]						×	×			×	×	×					
LOI et al. [LHVT17]	×					×					×						
BENEŠ et al. [BŠMM11]	×					×		×			×	×					
TALTON et al. [TLL*11]	×					×			×		×	×					
RITCHIE et al. [RMGH15]	×					×			×		×	×					
CHEN et al. [CZX*16]		×				×					×	×			×		
VECTOR FIELDS																	
IJIRI et al. [IMIM08]				×		×	×				×	×		×	×		
LI et al. [LBZ*11]	×					×					×				×		
SAPUTRA et al. [SKAM17]		×			×					×	×	×			×		
CURVES, SKETCHES & PAINTING																	
As Basis Elements																	
ANDERSON et al. [AW08]					×	×	×					×	×			×	
CHEN et al. [CSC12]													×				
XU et al. [XM09]		×					×				×		×				
MERRELL et al. [MM10]					×												×
ZEHNDER et al. [ZCT16]					×		×	×			×					×	×
As Control Mechanism																	
MĚCH et al. [MM12]*	×					×	×			×	×	×	×	×			
JACOBS et al. [JBMR18]		×			×	×	×	×						×		×	
LU et al. [LBW*14]		×			×	×				×			×				
ZHOU et al. [ZJL14]		×			×	×	×				×		×				
KAZI et al. [KIZD12]						×	×			×	×		×	×	×	×	
XING et al. [XCW14]						×	×	×		×	×		×	×	×	×	
ELEMENT PLACEMENT																	
YEH et al. [YM09]				×										×		×	×
GUERRERO et al. [GBLM16]	×			×				×								×	×

Table 2: *Control mechanisms in the state of the art for pattern generation designs.*

some way connected to the cloud-based storage of assets, settings and results; therefore, they function online and are easily shared. Collaboration is closely connected to the previously discussed issue of navigation histories. This is not only relevant for individual work processes but also for more general production pipelines in a commercial context. In this regard, the sharing and collaborative work on iterations, which involves multiple persons referencing different versions, is essential. Some work has been done (e.g., [SSTP15; OWL*18]) but further investigations of collaboration for creative control are called for.

As discussed in Section 6, control techniques are closely interrelated with the representation of the underlying models. Therefore, a more unified development of models across research communities would be beneficial. Expert knowledge of usability should especially be considered. However, further automation for the creation of complex decorative models also poses interesting challenges, such as abstraction [NSX*11], symmetry computation [CO11] and design space variations.

8.0.1.1. Machine Learning To integrate a machine learning framework into a procedural system is a fairly novel development. In the context of creative pattern generation, a summarization of the state of the art is not yet representative. The following describes some relevant work.

PHAN et al. [PLA*16] offer a data-driven recommendation system for circular ornamentation, employing a learned style and composition feature vector. Based on a custom ring-based layout system that represents, for example, plates, vases and a first decorative element chosen by the artist, the system completes a design. The artist can also chose to incrementally add elements manually, while the system accompanies this by suggesting suitable elements and placements. This work indicates the promising direction of using learned characteristics to further stimulating tools, which, for example, generate meaningful design suggestions.

RITCHIE et al. [RTHG16] make use of machine learning to improve the performance of the image-matching grammar-based models of RITCHIE et al. [RMGH15]. The updated system increases performance up to 10 times by integrating a neural network and sampling a learned constraint-satisfying approximation. Reported performances are overall below 3 seconds. Interactive performance is the foundation of all creative control means and hence of great importance.

The procedural content generation (PCG) for games community is pushing the general integration of artificial intelligence (AI) into a procedural creation process. A new paradigm of *mixed-initiative creative interfaces* is rising and is actively fostered, as an ACM Conference on Human Factors in Computing Systems (CHI) workshop under the same name in 2017 shows [DHF*17]. As the workshop summary states, it is the goal to "put human and computer in a tight interactive loop where each suggests, produces, evaluates, modifies, and selects creative outputs in response to the other." In order to achieve this, AI enables computer agency, and novel interfaces enable collaboration between computers and human users. The workshop brought PCG and interaction design researchers together, stressing the importance of bridging disciplines. Similarly to games, the context of creative pattern generation also constitutes a challenging but fruitful testing ground for the investigation of mixed-initiative creative interfaces and for the task of balancing artist control and automation, as this survey shows. In general, the involvement of the computer graphics community with its various topics and rich algorithmic knowledge would be promising.

8.0.1.2. Semantic Attributes The usage of semantic attributes presents a highly intuitive navigation technique, which so far has been successfully applied in the context of shape modifications, for example by YUMER et al. [YCHK15].

In the context of complex patterns, procedural textures constitute the most related field of investigation. For the control of procedural textures, methods are based on the analysis and description of texture in regard to human perception, which has a long research tradition. In his influential work, JULESZ [Jul81] defines textons as the basic units of pre-attentive human texture perception. Since then, this line of research has continued, and texture descriptions with perceptual [LDC*15] and semantic [MNN13; CMK*14] attributes have been investigated. DONG et al. [DWLS17] and LIU et al. [LGD*18] employed such features in first experiments for the navigation of a procedural texture space and for the generation of suitable textures by given features. However, the results of such studies are still limited and of varying quality - and the authors themselves [LGD*18] call their results experimental.

Nonetheless, these works present an interesting approach that is worth further investigation. Because many pattern designs are structured and follow an internal logic, it seems feasible to come up with a collection of suitable attributes. For example an ornamental design space is much smaller in comparison to all "textures in the wild" [CMK*14], and ornamentation could constitute a valuable context for further investigations into the incorporation of semantic attributes into a creative creation process.

8.1. Conclusions

Note: This is still a bit messy.

The overall challenge addressed in this survey is how to support artists in their work with meaningful control mechanisms. The investigation of controllability is put into the context of two-dimensional creative pattern designs. Procedural models and the computation of designs offer novel approaches to create content and benefits over traditional manual manufacturing. However, to provide control mechanisms that are intuitive to use and allow for individual designs is an ongoing research challenge.

To tackle this challenge, this survey provides first a better understanding of the creation process of an artist, means for creativity, and how to relate the identified characteristics to control mechanisms for digital tools. We dissect a creation process into overall characteristics and classify specific control mechanisms by their interaction types.

The analysis of the state of the art shows the capabilities of the different control mechanisms and potential trade-offs between approaches. For handling the ill-defined topic of creativity, we follow the definition of creativity as intentionally producing a novel and

surprising product. We discuss means for creative control and relate specific control mechanisms to creative processes. By this we further a more objective judging of the ability of a technique to support creativity and a detailed comparison of methods. However, several aspects of the analysis still leave room for interpretation. Knowledge from other disciplines, for example in regard to the perception of visual features, might be able to contribute with valuable insights. We hope that our results inspire such research towards a quantifiable analysis of creative control.

While the focus of this state of the art is on procedural models, various relevant data-driven approaches are integrated and work that solely specializes on creative designs highlighted. Our analysis shows that current work mainly focuses on specific and separated single aspects, which can not support overall creative creation processes. For more complete and meaningful solutions aspects of both, data-driven and procedural techniques are needed and must be merged to a unified whole.

References

[ABS06] ARBRUZZO, EMILY, BRISEN, ALEXANDER, and SOLOMAN, JONATHAN D. *Decoration 306090*. Vol. 10. Princeton Architectural Press, 2006 8, 9.

[AGYS14] ABDRASHITOV, RINAT, GUY, EMILIE, YAO, JIAXIAN, and SINGH, KARAN. "Mosaic: Sketch-based Interface for Creating Digital Decorative Mosaics". *Proceedings of the Joint Symposium on Computational Aesthetics, Non-Photorealistic Animation and Rendering, and Sketch-Based Interfaces and Modeling*. ACM, 2014, 5–10 18.

[APS08] ANASTACIO, F., PRUSINKIEWICZ, P., and SOUSA, M. C. "Sketch-based Parameterization of L-systems Using Illustration-inspired Construction Lines". *Proceedings of the Eurographics Conference on Sketch-Based Interfaces and Modeling*. Eurographics Association, 2008, 119–126. URL: http://dx.doi.org/10.2312/SBM/SBM08/119-126 17.

[AW08] ANDERSON, DUSTIN and WOOD, ZOË. "User driven two-dimensional computer-generated ornamentation". *Advances in Visual Computing*. Springer, 2008, 604–613 17, 19.

[BBT*06] BARLA, PASCAL, BRESLAV, SIMON, THOLLOT, JOËLLE, et al. "Stroke Pattern Analysis and Synthesis". *Computer Graphics Forum* 25.3 (2006), 663–671 14, 19.

[BD04] BOURQUE, ERIC and DUDEK, GREGORY. "Procedural Texture Matching and Transformation". *Computer Graphics Forum* 23.3 (2004), 461–468 2, 10, 14, 19.

[BDH14] BISKJAER, MICHAEL, DALSGAARD, PETER, and HALSKOV, KIM. "A Constraint-based Understanding of Design Spaces". *Proceedings of the Conference on Designing Interactive Systems*. ACM, 2014, 453–462 7.

[Bod04] BODEN, MARGARET A. *The Creative Mind: Myths and Mechanisms*. Routledge, 2004 6.

[Bod10] BODEN, MARGARET A. *Creativity and Art: Three Roads to Surprise*. Oxford University Press, 2010 6, 7.

[BŠMM11] BENEŠ, B., ŠT'AVA, O., MĚCH, R., and MILLER, G. "Guided Procedural Modeling". *Computer Graphics Forum* 30.2 (2011), 325–334 2, 10, 16, 19.

[BWCS14] BENEDETTI, LUCA, WINNEMÖLLER, HOLGER, CORSINI, MASSIMILIANO, and SCOPIGNO, ROBERTO. "Painting with Bob: Assisted Creativity for Novices". *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM, 2014, 419–428 7.

[CEW*08] CHEN, GUONING, ESCH, GREGORY, WONKA, PETER, et al. "Interactive Procedural Street Modeling". *ACM Transactions on Graphics* 27.3 (2008), 103:1–103:10 16.

[CFA16] CHEN, YILAN, FU, HONGBO, and AU, KIN CHUNG. "A Multi-level Sketch-based Interface for Decorative Pattern Exploration". *SIGGRAPH Asia Technical Briefs*. ACM, 2016, 26:1–26:4 18.

[CL14] CHERRY, ERIN and LATULIPE, CELINE. "Quantifying the Creativity Support of Digital Tools Through the Creativity Support Index". *ACM Transactions on Computer-Human Interaction* 21.4 (2014), 21:1–21:25 6, 7.

[CMK*14] CIMPOI, M., MAJI, S., KOKKINOS, I., et al. "Describing Textures in the Wild". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014 20.

[CNX*08] CHEN, XUEJIN, NEUBERT, BORIS, XU, YING-QING, et al. "Sketch-based Tree Modeling Using Markov Random Field". *ACM Transactions on Graphics* 27.5 (2008), 109:1–109:9 17.

[CO11] CULLEN, B. and O'SULLIVAN, C. "Symmetry Hybrids". *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*. ACM, 2011, 33–38 20.

[Cro93] CROMWELL, PETER R. "Celtic knotwork: Mathematical art". *The Mathematical Intelligencer* 15.1 (1993), 36–47 11.

[CSC12] CHEN, YU-SHENG, SHIE, JIE, and CHEN, LIEU-HEN. "A NPR System for Generating Floral Patterns based on L-System". *Bulletin of Networking, Computing, Systems, and Software* 1.1 (2012) 11, 17, 19.

[CYZL14] CHI, MING-TE, YAO, CHIH-YUAN, ZHANG, EUGENE, and LEE, TONG-YEE. "Optical illusion shape texturing using repeated asymmetric patterns". *The Visual Computer* 30.6 (2014), 809–819 12.

[CZX*16] CHEN, WEIKAI, ZHANG, XIAOLONG, XIN, SHIQING, et al. "Synthesis of filigrees for digital fabrication". *ACM Transactions on Graphics* 35.4 (2016), 98 16, 19.

[De 85] DE BONO, EDWARD. *Six thinking hats*. Little Brown and Company, 1985 7.

[DHF*17] DETERDING, SEBASTIAN, HOOK, JONATHAN, FIEBRINK, REBECCA, et al. "Mixed-Initiative Creative Interfaces". *Proceedings of the CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2017, 628–635 6, 20.

[DKMI13] DIVERDI, S., KRISHNASWAMY, A., MĚCH, R., and ITO, D. "Painting with Polygons: A Procedural Watercolor Engine". *IEEE Transactions on Visualization and Computer Graphics* 19.5 (2013), 723–735 17.

[DLW81] DUNHAM, DOUGLAS, LINDGREN, JOHN, and WITTE, DAVID. "Creating Repeating Hyperbolic Patterns". *SIGGRAPH Computer Graphics* 15.3 (1981), 215–223 12.

[Dre75] DRESSER, CHRISTOPHER. *Principles of decorative design*. Cassell Petter and Galpin, 1875 8.

[DS13] DOYLE, RICHARD BRIAN and SEMWAL, SUDHANSHU K. "Computational Celtic Canvas for Zoomorphs and Knotworks". *Proceedings of the 21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. 2013 11.

[DWLS17] DONG, JUNYU, WANG, LINA, LIU, JUN, and SUN, XIN. "A Procedural Texture Generation Framework Based on Semantic Descriptions". (2017). arXiv:1704.04141 [cs.CV] 20.

[EMP*02] EBERT, DAVID S., MUSGRAVE, F. KENTON, PEACHEY, DARWYN, et al. *Texturing and Modeling: A Procedural Approach*. 3rd. Morgan Kaufmann Publishers Inc., 2002 10.

[ESP08] ETEMAD, KATAYOON, SAMAVATI, FARAMARZ F., and PRUSINKIEWICZ, PRZEMYSLAW. "Animating Persian Floral Patterns". *Proceedings of the Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*. Eurographics Association, 2008, 25–32 11.

[EVC*15] EMILIEN, ARNAUD, VIMONT, ULYSSE, CANI, MARIE-PAULE, et al. "WorldBrush: Interactive Example-based Synthesis of Procedural Virtual Worlds". *ACM Transactions on Graphics* 34.4 (2015), 106:1–106:11 17.

[Gau10] GAUT, BERYS. "The Philosophy of Creativity". *Philosophy Compass* 5.12 (2010), 1034–1046 6.

[GBLM16] GUERRERO, PAUL, BERNSTEIN, GILBERT, LI, WILMOT, and MITRA, NILOY J. "PATEX: Exploring Pattern Variations". *ACM Transactions on Graphics* 35.4 (2016), 48:1–48:13 18, 19.

[GD10] GILET, G. and DISCHLER, J-M. "An Image-Based Approach for Stochastic Volumetric and Procedural Details". *Computer Graphics Forum* 29.4 (2010), 1411–1419 2, 10, 13, 19.

[Gda17] GDAWIEC, KRZYSZTOF. "Procedural Generation of Aesthetic Patterns from Dynamics and Iteration Processes". *International Journal of Applied Mathematics and Computer Science* 27.4 (2017), 827–837 12.

[GDG12a] GILET, G., DISCHLER, J-M., and GHAZANFARPOUR, D. "Multiple kernels noise for improved procedural texturing". *The Visual Computer* 28.6 (2012), 679–689 13, 14, 19.

[GDG12b] GILET, G., DISCHLER, J-M., and GHAZANFARPOUR, D. "Multi-scale Assemblage for Procedural Texturing". *Computer Graphics Forum* 31.7 (2012), 2117–2126 13, 15, 19.

[GLLD12] GALERNE, BRUNO, LAGAE, ARES, LEFEBVRE, SYLVAIN, and DRETTAKIS, GEORGE. "Gabor Noise by Example". *ACM Transactions on Graphics* 31.4 (2012), 73:1–73:9 13, 15, 19.

[GLM17] GALERNE, B., LECLAIRE, A., and MOISAN, L. "Texton Noise". *Computer Graphics Forum* (2017) 12, 13, 19.

[GSDC17] GUINGO, GEOFFREY, SAUVAGE, BASILE, DISCHLER, JEAN-MICHEL, and CANI, MARIE-PAULE. "Bi-Layer textures: a Model for Synthesis and Deformation of Composite Textures". *Computer Graphics Forum* 36.4 (2017), 111–122 13, 19.

[GSK12] GULATI, VISHAL, SINGH, KULWANT, and KATYAL, PUNEET. "A CAD Paradigm for Generating Woodworking Motifs". 47 (2012), 38–40 12.

[GSV*14] GILET, GUILLAUME, SAUVAGE, BASILE, VANHOEY, KENNETH, et al. "Local Random-phase Noise for Procedural Texturing". *ACM Transactions on Graphics* 33.6 (2014), 195:1–195:11 13, 19.

[GTS10] GULATI, VISHAL, TANDON, PUNEET, and SINGH, HARI. "A CAD Paradigm to Produce Zillij Style of Geometrical Patterns for Wooden Carvings". *International Journal of Computer Applications* 3.3 (2010), 28–35 12.

[HB95] HEEGER, DAVID J. and BERGEN, JAMES R. "Pyramid-based Texture Analysis/Synthesis". *Proceedings of the SIGGRAPH Conference on Computer Graphics and Interactive Techniques*. ACM, 1995, 229–238 11.

[HF04] HAVEMANN, S. and FELLNER, D. "Generative parametric design of Gothic window tracery". *Proceedings Shape Modeling Applications*. 2004, 350–353 12.

[HLT*09] HURTUT, T., LANDES, P.-E., THOLLOT, J., et al. "Appearance-guided Synthesis of Element Arrangements by Example". *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2009, 51–60 14, 19.

[HMVI13] HENDRIKX, MARK, MEIJER, SEBASTIAAN, VAN DER VELDEN, JOERI, and IOSUP, ALEXANDRU. "Procedural Content Generation for Games: A Survey". *ACM Transactions on Multimedia Computing, Communications, and Applications* 9.1 (2013), 1:1–1:22 2, 10, 11.

[HS12] HAMEKASI, NADER and SAMAVATI, FARAMARZ. "Designing Persian Floral Patterns using Circle Packing". *Proceedings of the International Conference on Computer Graphics Theory and Applications*. 2012, 135–142 11.

[Iga10] IGARASHI, YUKI. "DECO: A Designing Editor for Line Stone Decoration". *ACM SIGGRAPH Posters*. ACM, 2010, 34:1–34:1 18.

[IMIM08] IJIRI, TAKASHI, MĚCH, RADOMÍR, IGARASHI, TAKEO, and MILLER, GAVIN. "An Example-based Procedural System for Element Arrangement". *Computer Graphics Forum* 27.2 (2008), 429–436 11, 14–16, 19.

[Ise16] ISENBERG, TOBIAS. "Interactive NPAR: What type of tools should we create?": *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering*. 2016, 89–96 6.

[JBMR18] JACOBS, JENNIFER, BRANDT, JOEL R., MĚCH, RADOMÍR, and RESNICK, MITCHEL. "Dynamic Brushes: Extending Manual Drawing Practices with Artist-Centric Programming Tools". *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. ACM, 2018, D316:1–D316:4 17, 19.

[Jul81] JULESZ, BÉLA. "Textons, the elements of texture perception, and their interactions". *Nature* 290 (1981), 91–97 20.

[Kap02] KAPLAN, CRAIG STEVEN. "Computer Graphics and Geometric Ornamental Design". University of Washington. PhD thesis. 2002 11.

[KB18] KHAMJANE, AZIZ and BENSLIMANE, RACHID. "Generating Islamic Quasi-Periodic Patterns: A New Method". *ACM Journal on Computing and Cultural Heritage* 11.3 (2018), 13:1–13:18 11.

[KC03] KAPLAN, MATTHEW and COHEN, ELAINE. "Computer Generated Celtic Design". *Proceedings of the Eurographics Workshop on Rendering*. Eurographics Association, 2003, 9–19 11.

[KH17] KANG, HYEONGYEOP and HAN, JUNGHYUN. "Feature-preserving Procedural Texture". *The Visual Computer* 33.6-8 (2017), 761–768 13, 19.

[KIZD12] KAZI, RUBAIAT HABIB, IGARASHI, TAKEO, ZHAO, SHENGDONG, and DAVIS, RICHARD. "Vignette: Interactive Texture Design and Manipulation with Freeform Gestures for Pen-and-ink Illustration". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, 1727–1736 17–19.

[KS04] KAPLAN, CRAIG S. and SALESIN, DAVID H. "Islamic Star Patterns in Absolute Geometry". *ACM Transactions on Graphics* 23.2 (2004), 97–119 11, 12.

[LBW*14] LU, JINGWAN, BARNES, CONNELLY, WAN, CONNIE, et al. "DecoBrush: Drawing Structured Decorative Patterns by Example". *ACM Transactions on Graphics* 33.4 (2014), 90:1–90:9 17, 19.

[LBZ*11] LI, YUANYUAN, BAO, FAN, ZHANG, EUGENE, et al. "Geometry Synthesis on Surfaces Using Field-Guided Shape Grammars". *IEEE Transactions on Visualization and Computer Graphics* 17.2 (2011), 231–243 11, 16, 19.

[LDC*15] LIU, JUN, DONG, JUNYU, CAI, XIAOXU, et al. "Visual Perception of Procedural Textures: Identifying Perceptual Dimensions and Predicting Generation Models". (2015). PLoS ONE 10(6): e0130335 20.

[LFB*13] LUKÁČ, MICHAL, FIŠER, JAKUB, BAZIN, JEAN-CHARLES, et al. "Painting by Feature: Texture Boundaries for Example-based Image Creation". *ACM Transactions on Graphics* 32.4 (2013), 116:1–116:8 18.

[LGD*18] LIU, JUN, GAN, YANHAI, DONG, JUNYU, et al. "Perception-driven procedural texture generation from examples". *Neurocomputing* 291 (2018), 21–34 20.

[LHVT17] LOI, HUGO, HURTUT, THOMAS, VERGNE, ROMAIN, and THOLLOT, JOELLE. "Programmable 2D Arrangements for Element Texture Design". *ACM Transactions on Graphics* 36.3 (2017), 27:1–27:17 15, 19.

[LHW*06] LIN, WEN-CHIEH, HAYS, J., WU, CHENYU, et al. "Quantitative Evaluation of Near Regular Texture Synthesis Algorithms". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2006, 427–434 2.

[LLC*10] LAGAE, A., LEFEBVRE, S., COOK, R., et al. "State of the Art in Procedural Noise Functions". *Proceedings of the Conference of the European Association for Computer Graphics*. The Eurographics Association, 2010 2, 10, 12.

[LLD12a] LASRAM, ANASS, LEFEBVRE, SYLVAIN, and DAMEZ, CYRILLE. "Procedural Texture Preview". *Computer Graphics Forum* 31.2 (2012), 413–420 2, 10.

[LLD12b] LASRAM, ANASS, LEFEBVRE, SYLVAIN, and DAMEZ, CYRILLE. "Scented Sliders for Procedural Textures". *Proceedings of the Conference of the European Association for Computer Graphics*. The Eurographics Association, 2012 2, 10.

[LP00] LEFEBVRE, LAURENT and POULIN, PIERRE. "Analysis and Synthesis of Structural Textures". *Proceedings of the Graphics Interface Conference*. 2000, 77–86 13, 19.

[LVLD10] LAGAE, ARES, VANGORP, PETER, LENAERTS, TOON, and DUTRÉ, PHILIP. "Procedural Isotropic Stochastic Textures by Example". *Computers and Graphics* 34.4 (2010), 312–321 2, 10, 12, 19.

[MBS*11] MAHARIK, RON, BESSMELTSEV, MIKHAIL, SHEFFER, ALLA, et al. "Digital Micrography". *ACM Transactions on Graphics* 30.4 (2011), 100:1–100:12 16.

[MM10] MERRELL, PAUL and MANOCHA, DINESH. "Example-based Curve Synthesis". *Computers and Graphics* 34.4 (2010), 304–311 17, 19.

[MM12] MĚCH, RADOMÍR and MILLER, GAVIN. "The Deco framework for interactive procedural modeling". *Journal of Computer Graphics Techniques* 1.1 (2012), 43–99. (Visited on 12/03/2015) 10, 17–19.

[MNN13] MATTHEWS, T., NIXON, M. S., and NIRANJAN, M. "Enriching Texture Analysis with Semantic Data". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, 1248–1255 20.

[MOT99] MOUGHTIN, CLIFF, OC, TANER, and TIESDELL, STEVEN. *Urban Design: Ornament and Decoration*. Architectural Press, 1999 8, 9.

[MW09] MARKMAN, A.B. and WOOD, K.L. *Tools for Innovation: The Science Behind the Practical Methods That Drive New Ideas*. Oxford University Press, 2009 7.

[MWLT13] MA, CHONGYANG, WEI, LI-YI, LEFEBVRE, SYLVAIN, and TONG, XIN. "Dynamic Element Textures". *ACM Transactions on Graphics* 32.4 (2013), 90:1–90:10 14, 15.

[MWT11] MA, CHONGYANG, WEI, LI-YI, and TONG, XIN. "Discrete Element Textures". *ACM Transactions on Graphics* 30.4 (2011), 62:1–62:10 14, 19.

[NSX*11] NAN, LIANGLIANG, SHARF, ANDREI, XIE, KE, et al. "Conjoining Gestalt Rules for Abstraction of Architectural Drawings". *ACM Transactions on Graphics* 30.6 (2011), 185:1–185:10 20.

[Ost98] OSTROMOUKHOV, VICTOR. "Mathematical tools for computer-generated ornamental patterns". Springer, 1998, 193–223 11.

[OW10] ONARHEIM, BALDER and WILTSCHNIG, STEFAN. "Opening and Constraining: Constraints and Their Role in Creative Processes". *Proceedings of the DESIRE Network Conference on Creativity and Innovation in Design*. Desire Network, 2010, 83–89 7.

[OWL*18] O'LEARY, JASPER, WINNEMÖLLER, HOLGER, LI, WILMOT, et al. "Charrette: Supporting In-Person Discussions Around Iterations in User Interface Design". *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2018, 535:1–535:11 20.

[Oxf17] OXFORD ENGLISH DICTIONARY ONLINE. http://www.oed.com. Accessed August 20, 2017. 2017 2, 8.

[OZH15] OUYANG, P., ZHAO, W., and HUANG, X. "Beautiful Math, Part 5: Colorful Archimedean Tilings from Dynamical Systems". *IEEE Computer Graphics and Applications* 35.6 (2015), 90–96 12.

[Per85] PERLIN, KEN. "An Image Synthesizer". Vol. 19. 3. ACM, 1985, 287–296 10.

[PGDG16] PAVIE, NICOLAS, GILET, GUILLAUME, DISCHLER, JEAN-MICHEL, and GHAZANFARPOUR, DJAMCHID. "Procedural texture synthesis by locally controlled spot noise". *Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. 2016 13, 19.

[PHL*09] PALUBICKI, WOJCIECH, HOREL, KIPP, LONGAY, STEVEN, et al. "Self-organizing Tree Models for Image Synthesis". *ACM Transactions on Graphics* 28.3 (2009), 58:1–58:10 17.

[PLA*16] PHAN, H. Q., LU, J., ASENTE, P., et al. "Patternista: Learning Element Style Compatibility and Spatial Composition for Ring-based Layout Decoration". *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*. Eurographics Association, 2016, 79–88 20.

[Pru90] PRUSINKIEWICZ, PRZEMYSLAW. *The algorithmic beauty of plants*. Springer, 1990 10.

[RMGH15] RITCHIE, DANIEL, MILDENHALL, BEN, GOODMAN, NOAH D., and HANRAHAN, PAT. "Controlling Procedural Modeling Programs with Stochastically-ordered Sequential Monte Carlo". *ACM Transactions on Graphics* 34.4 (2015), 105:1–105:11 10, 16, 19, 20.

[RTHG16] RITCHIE, DANIEL, THOMAS, ANNA, HANRAHAN, PAT, and GOODMAN, NOAH D. "Neurally-guided Procedural Models: Amortized Inference for Procedural Graphics Programs Using Neural Networks". *Proceedings of the International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2016, 622–630 20.

[Sal02] SALESIN, DAVID H. "Non-Photorealistic Animation and Rendering: 7 Grand Challenges". *Keynote talk at Second International Symposium on Non-Photorealistic Animation and Rendering*. 2002 6.

[ŠBM*10] ŠT'AVA, O., BENEŠ, B., MĚCH, R., et al. "Inverse Procedural Modeling by Automatic Generation of L-systems". *Computer Graphics Forum* 29.2 (2010), 665–674. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2009.01636.x/abstract (visited on 06/09/2015) 2, 10, 14, 19.

[Shn07] SHNEIDERMAN, BEN. "Creativity Support Tools: Accelerating Discovery and Innovation". *Communications of the ACM* 50.12 (2007), 20–32 6, 7.

[SKAM17] SAPUTRA, REZA ADHITYA, KAPLAN, CRAIG S., ASENTE, PAUL, and MĚCH, RADOMÍR. "FLOWPAK: Flow-based Ornamental Element Packing". *Proceedings of the Graphics Interface Conference*. Canadian Human-Computer Communications Society, 2017, 8–15 11, 16, 19.

[SLD17] SHUGRINA, MARIA, LU, JINGWAN, and DIVERDI, STEPHEN. "Playful Palette: An Interactive Parametric Color Mixer for Artists". *ACM Transactions on Graphics* 36.4 (2017), 61:1–61:10 6.

[SP16] SANTONI, CHRISTIAN and PELLACINI, FABIO. "gTangle: A Grammar for the Procedural Generation of Tangle Patterns". *ACM Transactions on Graphics* 35.6 (2016), 182:1–182:11 2, 15, 16, 19.

[SSG*17] SUMMERVILLE, ADAM, SNODGRASS, SAM, GUZDIAL, MATTHEW, et al. "Procedural Content Generation via Machine Learning". (2017). arXiv:1702.00539 [cs.AI] 11.

[SSTP15] SALVATI, GABRIELE, SANTONI, CHRISTIAN, TIBALDO, VALENTINA, and PELLACINI, FABIO. "MeshHisto: Collaborative Modeling by Sharing and Retargeting Editing Histories". *ACM Transactions on Graphics* 34.6 (2015), 205:1–205:10 20.

[STBB14] SMELIK, RUBEN M., TUTENEL, TIM, BIDARRA, RAFAEL, and BENES, BEDRICH. "A Survey on Procedural Modeling for Virtual Worlds". *Computer Graphics Forum* (2014) 2, 10.

[Sto05] STOKES, P.D. *Creativity from Constraints: The Psychology of Breakthrough*. Springer, 2005 7.

[SVO11] SHIH, PATRICK C., VENOLIA, GINA, and OLSON, GARY M. "Brainstorming Under Constraints: Why Software Developers Brainstorm in Groups". *Proceedings of the BCS Conference on Human-Computer Interaction*. British Computer Society, 2011, 74–83 7.

[Tak16] TAKAYAMA, JOE. "Medallions". *SIGGRAPH Asia 2016 Art Gallery*. ACM, 2016, 15:1–15:1 11.

[TLL*11] TALTON, JERRY O., LOU, YU, LESSER, STEVE, et al. "Metropolis procedural modeling". *ACM Transactions on Graphics* 30.2 (2011), 1–14 10, 15, 16, 19.

[TMNY04] TERRY, MICHAEL, MYNATT, ELIZABETH D., NAKAKOJI, KUMIYO, and YAMAMOTO, YASUHIRO. "Variation in Element and Action: Supporting Simultaneous Development of Alternative Solutions". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2004, 711–718 7.

[TWO16] TODI, KASHYAP, WEIR, DARYL, and OULASVIRTA, ANTTI. "Sketchplore: Sketch and Explore with a Layout Optimiser". *Proceedings of the ACM Conference on Designing Interactive Systems*. ACM, 2016, 543–555 18.

[TYK*12] TALTON, JERRY, YANG, LINGFENG, KUMAR, RANJITHA, et al. "Learning Design Patterns with Bayesian Grammar Induction". *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM, 2012, 63–74 14, 19.

[TYSB11] TOGELIUS, J., YANNAKAKIS, G. N., STANLEY, K. O., and BROWNE, C. "Search-Based Procedural Content Generation: A Taxonomy and Survey". *IEEE Transactions on Computational Intelligence and AI in Games* 3.3 (2011), 172–186 2.

[VAW*09] VANEGAS, CARLOS A., ALIAGA, DANIEL G., WONKA, PETER, et al. "Modeling the Appearance and Behavior of Urban Spaces". *Proceedings of the Conference of the European Association for Computer Graphics*. The Eurographics Association, 2009 2.

[Wal12] WALT DISNEY ANIMATION STUDIOS. *Paperman*. https://www.disneyanimation.com/studio/our-films. Accessed August 20, 2018. 2012 1.

[War96] WARD, JAMES. *The principles of ornament*. Chapman and Hall, 1896 8, 9.

[Wei06] WEISBERG, R.W. *Creativity: Understanding Innovation in Problem Solving, Science, Invention, and the Arts*. Wiley, 2006 6.

[Whi10] WHITEHEAD, JIM. "Toward Proccedural Decorative Ornamentation in Games". *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. ACM, 2010, 9:1–9:4 2, 11.

[WLKT09] WEI, LI-YI, LEFEBVRE, SYLVAIN, KWATRA, VIVEK, and TURK, GREG. "State of the art in example-based texture synthesis". *Proceedings of the Conference of the European Association for Computer Graphics*. The Eurographics Association, 2009, 93–117 11.

[Wor96] WORNUM, RALPH N. *Analysis of ornament*. Chapman and Hall, 1896 8.

[WZS98] WONG, MICHAEL T., ZONGKER, DOUGLAS E., and SALESIN, DAVID H. "Computer-generated Floral Ornament". *Proceedings of the Conference on Computer Graphics and Interactive Techniques*. ACM, 1998, 423–434 2, 8–11, 15, 17, 19.

[XCW14] XING, JUN, CHEN, HSIANG-TING, and WEI, LI-YI. "Autocomplete Painting Repetitions". *ACM Transactions on Graphics* 33.6 (2014), 172:1–172:11 18, 19.

[XHC*18] XIA, HAIJUN, HENRY RICHE, NATHALIE, CHEVALIER, FANNY, et al. "DataInk: Direct and Creative Data-Oriented Drawing". *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 2018, 223:1–223:13 18.

[XKG*16] XING, JUN, KAZI, RUBAIAT HABIB, GROSSMAN, TOVI, et al. "Energy-Brushes: Interactive Tools for Illustrating Stylized Elemental Dynamics". *Proceedings of the Symposium on User Interface Software and Technology*. ACM, 2016, 755–766 17.

[XM09] XU, LING and MOULD, DAVID. "Magnetic Curves: Curvature-Controlled Aesthetic Curves Using Magnetic Fields". *Proceedings of the Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*. The Eurographics Association, 2009 17, 19.

[XM15] XU, LING and MOULD, DAVID. "Procedural Tree Modeling with Guiding Vectors". *Computer Graphics Forum* 34.7 (2015), 47–56 16.

[YCHK15] YUMER, MEHMET ERSIN, CHAUDHURI, SIDDHARTHA, HODGINS, JESSICA K., and KARA, LEVENT BURAK. "Semantic Shape Editing Using Deformation Handles". *ACM Transactions on Graphics* 34.4 (2015), 86:1–86:12 20.

[YM09] YEH, YI-TING and MĚCH, RADOMÍR. "Detecting Symmetries and Curvilinear Arrangements in Vector Art". *Computer Graphics Forum* 28.2 (2009), 707–716. (Visited on 06/10/2015) 18, 19.

[ZCT16] ZEHNDER, JONAS, COROS, STELIAN, and THOMASZEWSKI, BERNHARD. "Designing Structurally-sound Ornamental Curve Networks". *ACM Transactions on Graphics* 35.4 (2016), 99:1–99:10 11, 17, 19.

[ZJL14] ZHOU, SHIZHE, JIANG, CHANGYUN, and LEFEBVRE, SYLVAIN. "Topology-constrained Synthesis of Vector Patterns". *ACM Transactions on Graphics* 33.6 (2014), 215:1–215:11 17, 19.