



Pacemod: parametric contour-based modifications for glyph generation

Dmitri Presnov¹ · Marlena Berels¹ · Andreas Kolb¹

Accepted: 22 July 2023 / Published online: 12 August 2023
© The Author(s) 2023

Abstract

Metaphoric glyphs are intuitively related to the underlying problem domain and enhance the readability and learnability of the resulting visualization. Their construction, however, implies an appropriate modification of the base icon, which is a predominantly manual process. In this paper, we introduce the *parametric contour-based modification (PACEMOD)* approach that lays the foundations of automated, controllable icon manipulations. Technically, the PACEMOD parametric representation utilizes diffusion curves, enriched with new degrees of freedom in arc-length parameterization, which allows for manipulation of the icon contours' geometry and the related color attributes. Moreover, we propose an implementation of our generic approach for a specific, automated design of metaphoric glyphs, based on periodic, wave-like contour modifications. Finally, the practicality of such periodic contour modifications is demonstrated by two visualization examples, which comprise uncertainty visualization of a rain forecast and gradient glyphs applied to COVID-19 data. In summary, with the PACEMOD approach we introduce an instrument that facilitates a user-centered design of metaphoric glyphs and provides a generic basis for potential further implementations according to specific applications.

Keywords Glyphs · Icon-based visualization · Contour modification · Visualization design · Visual encoding

1 Introduction

Icon-based or iconographic techniques, according to the taxonomy of Keim and Kriegel [31], form one fundamental class of methods for visual exploration of multivariate, multidimensional data. Their characteristic property is the mapping of data dimensions to varying visual features [16], i.e., visual variables or channels, such as size, color hue, luminance (or color value), grain, orientation and shape [3]. In literature, there is no clear consensus on differences between icons and glyphs. For the sake of consistency, in this paper we follow the definition that *icons* “represent a sign that itself resembles the qualities of the object it stands for”, while *glyphs* “rep-

resent different data variables by a set of visual channels” [7].

Glyph design involves two main tasks that affect the intuitive mapping of data variables to application-related visual variables, i.e., (1) the selection of an appropriate initial visual object, or icon, including shape and color, and (2) the modification of the visual variables of the initial icon, i.e., its geometry and related color attributes. To enhance the intuitivity, metaphoric glyphs make heavy use of familiar and well-understood visuo-spatial phenomena related to the underlying problem domain [43] in both tasks. The main advantages of metaphoric glyphs relate to their potential of increased readability in case of realistic glyphs [19], and improved data understanding [21], e.g., by mapping data to corresponding glyph parts [48].

On the technical level, the design of metaphoric glyphs is empowered and, at the same time, limited by the given means of manipulating the glyph's base icon and its visual variables, which often involves a large amount of manual graphics design. Examples are glyphs to visualize the health state of corn cobs [39], environmental data related to forest fires using leaf-like glyphs [22], or car glyphs that map car-related data to corresponding parts of the base icon [48]. More

✉ Dmitri Presnov
dmitri.presnov@uni-siegen.de

Marlena Berels
marlena.berels@student.uni-siegen.de

Andreas Kolb
andreas.kolb@uni-siegen.de

¹ Computer Graphics and Multimedia Systems Group,
University of Siegen, Siegen, Germany

automated approaches include procedural and data-driven methods. Procedural methods, such as RoseShape glyphs [8, 33] that are sinusoids plotted in polar coordinates applied to circles, have a very restrict set of base icons. Data-driven methods, such as the automatic generation of emoji-like metaphoric glyphs [14] are hardly controllable regarding the variation of their visual variable. In summary, there is a gap regarding automated and generalizable approaches for generation of metaphoric glyphs with user control, which this paper aims to fill.

In this paper, we introduce the PArametric Contour-basEd MODification (PACEMOD) concept, a novel approach that allows for controllable geometric and color modifications of an icon provided by the user. Our approach enhances the automated generation of metaphoric glyphs as it has two main technical features, particularly it (1) allows almost any base icon as input and (2) supports the design of various techniques for modification of the icon contour's geometry and the related color attributes. Technically, we utilize *diffusion curves* [40] as parametric representation of a user given icon, which we reparametrize to add new degrees of freedom (DOF) in an arc-length encoded manner. Using these DOFs allows us to modify the geometry and color of icon's contours in a consistent way, creating visual variables for data encoding but also retaining the icon's overall shape, and thus its recognizability. We exemplify our PACEMOD concept by implementing periodic wave-like shape and color modifications and size variations, optionally applied to selected icon parts. This implementation is far from being exhaustive, i.e., our concept is open to allow other modification approaches for the visual variables. In summary, our paper comprises the following contributions.

- **PACEMOD**, a new, diffusion-curve based concept of parametric, contour-based modification of icons, comprising
 - the parametrization of the icon contours as diffusion curves using B-splines,
 - the methodology to insert the DOFs in an arc-length encoded manner, required to control the visual variables, and
 - the utilization of distance transforms (DT) to allow further, geometry related contour manipulations and the prevention of self-intersection.
- The implementation of the PACEMOD concept with a focus on the automated, wave-like modification of the icon contour's shape and color.
- Two application examples, namely uncertainty visualization for rain forecast and gradient glyphs applied to COVID 19 data.

2 Prior work

We briefly discuss prior work, conceptually related to our PACEMOD concept.

Metaphoric glyphs Metaphoric glyphs form a specific sub-group of glyphs that try to enhance the underlying communication process by additionally utilizing visual analogies from the related application domain and ultimately strive for “the picture becomes the thing it represents” [43]. Several works underline their potential to improve readability [21]. For a general overview of glyph design and application, we refer the reader to the surveys from Ward [49], Borgo et al. [7] and Fuchs et al. [21].

Technically, approaches generate metaphoric glyphs are largely dominated by manual design processes or rely on structured databases. Nocke et al. [39], for example, propose a mosaic paradigm that decomposes the icon into tiles, alters the tiles in size, shape or color according to the data values, and recombines them to achieve the final corn glyph. Fuchs et al. [22] propose a manual design of leaf-shaped glyphs to utilize the humans ability to visually discriminate natural shapes by modifying, e.g., the leaf's morphology, venation and boundary for visualizing multidimensional data related to environmental events such as forest fires. Often, icons are combined with abstract glyph components. Legg et al. [32] and Chung et al. [12], for example, propose wave-like shape deformations and color modifications of a circle to encode a sportsman's performance, which surrounds the icon describing the specific sports event to be assessed. Li et al. [34] propose a dashboard-like glyph to summarize the key features of locations for billboard selection, which combines abstract glyphs and feature-related icons.

Besides glyphs based on parametric shape representations, such as RoseShape, i.e., polar sinusoidal plots resulting in a flower-like glyph [8], different data-driven approaches have been developed. Cunha et al. [14], for example, present a data-driven strategy for the automatic generation of emoji-like metaphoric glyphs utilizing a structured emojiing database [13]. Ying et al. [53] presented GlyphCreator, a tool for deep learning based decomposition of circle-like abstract glyphs into several visual elements, which are then manually bound to the input data attributes.

Contour-based methods Contour-based uncertainty visualization methods [6] relate to the general question of how to modify shape contours, also addressed by our approach.

Contour-based approaches to encode uncertainty in the underlying data include the variation of the contour lines' width [1], the usage of sets of contours in regions with high uncertainty in segmentation [41], and the modification of graph nodes and radial color gradients to encode the relation uncertainties in graphs. Görtler et al. [24] propose bubble

treemaps as an extension of circular treemaps that encode uncertainty using wave-like modifications and blur-effects applied to the circular-arc spline contours during the treemap generation process. Holliman et al. [26] use abstract circle-shaped glyphs for uncertainty visualization with wave-like modified contours, manually modeled in blender. Both techniques are intrinsically linked to a specific primitive as the glyph’s base shape, i.e., a circle.

A further group of established multivariate visualization approaches are radial methods such as star plots or pie charts (for an overview see, e.g., [17]), which are indirectly related to contour modifications. However, being based on a regular radial shape with a defined center and radius, they are hardly applicable to arbitrary iconic shapes and thus are less suitable for generation of metaphoric glyphs.

Diffusion Curves Vector-based image representations allow a more flexible handling of arbitrary shapes. Based on early work of Elder and Goldberg [18], diffusion curves image (DCI) has been proposed by Orzan et al. [40]. DCIs are a widely used vector graphics form that combines benefits of raster and vector-based representations, for which various functionalities and improvements have been developed over the years. The main focus lies on the conversion of raster images to DCI [36], enhanced DCI representation [52], DCI editing [27, 29, 35], and efficient rendering of DCIs [28, 47]. The editing and manipulation approaches proposed so-far for DCI are all based on manual intervention. Examples include the methods of Jeschke and colleagues [27, 29], who propose a click-and-drag metaphor for manipulating diffusion curve properties, and Lu et al. [35], who present a combination of global and local deformations for manipulating coarse and fine image content, respectively. In summary, there exists a rich tool set for utilizing DCI, but there is no adequate approach to control an icon’s visual variables as required for automated glyph generation.

PACEMOD in the context of prior work Our PACEMOD concept is conceptually related to the aforementioned contour-based techniques, especially to [24, 26] in using wave-like contour modifications. However, the prior approaches focus on a specific visualization tasks, e.g., uncertainty visualization, and use a specific base shape, yielding an appropriate, but relatively limited design space. Moreover, automated glyph generation is mainly realized for simple base geometries like circles [24, 26], while glyphs with more complex base shapes are commonly designed manually [22]. In contrast, we aim at a *generic glyph generation concept* comprising technical functionalities for a *flexible and automated* appearance modification of an *arbitrary iconic visual object*, used as base shape. Thus, the present concept can be potentially used in various application domains with little to no adaptation efforts. Technically, this is achieved using a

DCI-based parametric glyph representation along with a set of contour modification rules and respective control parameters.

3 PACEMOD concept

This section describes the technical foundations of the proposed approach for PARAMetric Contour-basEd MODifications (PACEMOD) of an icon. The discussion of the basic principles comprises the representation of the initial icon (Sect. 3.1) and the pre-processing (Sect. 3.2), the general approach to contour-based modifications (Sect. 3.3), and the final post-processing (Sect. 3.4). A specific approach implementation for periodic, wave-like modifications of the contour’s geometry and color is explained in Sect. 4. An overview of the PACEMOD generation process is represented in Fig. 1. The main notations, used for the PACEMOD description, are provided in Table 1.

3.1 Icon representation

To achieve controllable and automatable contour modifications, our approach uses the *parametric* diffusion curve image (DCI) representation [40] of the given base icon. A DCI represents an image as a set of K cubic Bézier curves in conjunction with *color parameters*, i.e., color values and blur attributes located at parametric position u along the respective Bézier curves (our current PACEMOD concept omits the blur parameter due to its visual dependency on color attributes). These curves are defined via Bézier control points $\{\mathbf{P}_i\}_{i=0}^{4,K} \subset \mathbb{R}^2$ and color control points $\{\mathbf{C}_i^l(u)\}_{i=0}^M, \{\mathbf{C}_i^r(u)\}_{i=0}^N, u \in [0, K]$ (see also Fig. 2, step 2a–b).

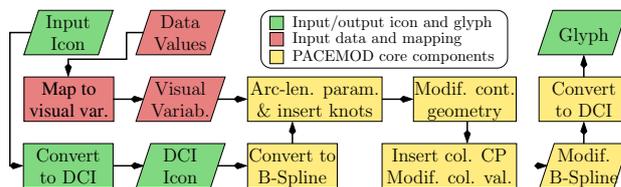


Fig. 1 The PACEMOD concept applied to glyph design

Table 1 Table of symbols (cp = control point)

\mathbf{P}_i	Orig. Bezier cp	K	# Bezier curves
$\mathbf{C}_i^{l/r}$	Orig. le./ri. col. cp	$\tilde{\mathbf{C}}_i^{l/r}$	Modif. le./ri. col. cp
\mathbf{D}_i	Orig. B-spline cp	$\tilde{\mathbf{D}}_i$	Modif. B-spline cp
t, \mathbf{T}	Orig. knot (vector)	$\tilde{t}, \tilde{\mathbf{T}}$	Modif. knot (vector)
u	Orig. color param	\tilde{u}	Modif. color param

We assume that the given base icon is composed of one or several closed, none-overlapping contours and that the regions defined by these contours have a homogeneous inner color, since a color gradient strongly restricts the options of an automated color modification.

3.2 Pre-processing

We use the diffusion curves drawing tool from Orzan et al. [40] to convert the given raster icon into a DCI. As modifications of the individual Bèzier segments would easily lead to unwanted cracks, we convert the DCI Bèzier representation into cubic B-spline curves. To increase the curve continuity wherever possible, we construct C^1 -transitions if consecutive Bèzier segments have first derivatives with (approximately) the same direction. Deviations in tangent length are corrected by an appropriate interval scaling, while slight directional deviation below a threshold α (we used $\alpha = 3.5^\circ$) are ignored.

More precisely, given the K (consecutive) Bèzier curves $b^i(u)$, $i = 0, \dots, K - 1$ with control points $\mathcal{P}^i = \{\mathbf{P}_{3i}, \dots, \mathbf{P}_{3i+3}\}$, we convert the i -th Bèzier segment by adapting the parameter interval to fit the tangent length of the end point of the prior segment. In case the directional deviation is below the threshold α , the transition is assumed to be C^1 , and we append a double knot to the initial B-spline knot vector \mathbf{T} and drop the last de Boor point from the control point list \mathbf{D} . In case of a C^0 transition, a 3-folded knot is

appended (see Fig. 2a). Subsequently, the parametric positions of color attributes are transformed by mapping them into interval defined by \mathbf{T} .

The resulting PACEMOD icon representation comprises de Boor points $\{\mathbf{D}_i\}_{i=0}^L$ with the corresponding knot vector $\mathbf{T} = \{t_i\}_{i=0}^{L+4}$, and color control points $\{\mathbf{C}_i^l(u')\}_{i=0}^M$ and $\{\mathbf{C}_i^r(u')\}_{i=0}^N$, where u' refers to the color parameter after the conversion to B-splines (see also Fig. 2, step 3a–b).

3.3 General contour-based modification

According to the two different types of the PACEMOD control parameters, icon's geometry and color can be modified separately.

3.3.1 Geometry

Geometric modifications are accomplished by a direct manipulation, i.e., translation, of the contour points \mathbf{Q} belonging to the corresponding B-Spline and the subsequent interpolation. They are controlled by source position on the curve and the translation vectors. In the following, the prerequisites and basic functionalities, allowing such modifications, are considered.

Arc-length parametrization and knot vector adjustment. In general, defining the source position in terms of arc length

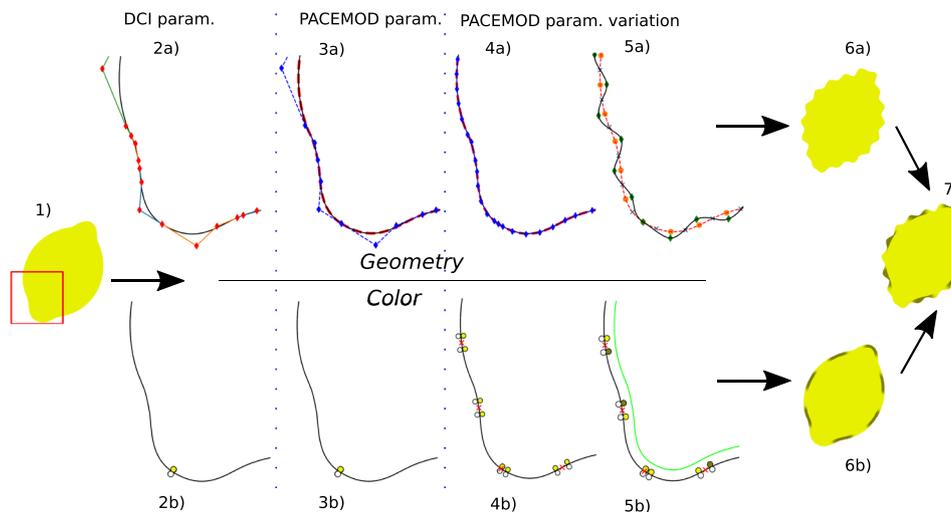


Fig. 2 The PACEMOD modifications: The input raster image (1) is converted into a DCI, comprising Bèzier control points for geometry (2a) and color control points (2b). Afterward, the geometry is converted into B-splines and a C^1 approximation is applied (3a) w/o changing the color control points (3b). In the next step, DOFs are added according to the target knot point and color border positions, resulting in an arc-length re-parametrization (4a) and new color points at the virtual borders of color intervals (4b: red crosses), respectively. The final shape

modification transforms subgroups of knot points (5a: orange points are moved to green points; blue crosses remain unchanged). The color modification changes the color points of each second interval (5b: darker circles) at the icon's inside, while the outer color is masked out and remains; a diffusion barrier (light green) is added to limit the diffusion and maintain the inner color unchanged. The final glyph is obtained by re-conversion into standard DCI and rendering (6a, b; 7)

s instead of using the original parametric space provides a more intuitive control. An arc-length parametrization of a B-Spline $b(s)$ is realized by means of a numerically calculated lookup table.

Moreover, based on $b(s)$, the knot vector \mathbf{T} is converted to $\tilde{\mathbf{T}}$ by shifting the original knots or inserting new ones, according to the following requirements. For a better modification control, the curve points \mathbf{Q} to be translated are converted to *knot points* [15], i.e., $\mathbf{Q} = b(\tilde{t})$ with $\tilde{t} \in \tilde{\mathbf{T}}$. The B-splines created in the pre-processing step (see Sect. 3.2) do not always have enough DOFs to interpolate the translated curve points. In such cases, additional DOFs are created inserting further knots by means of Boehm’s algorithm [5]. Furthermore, additional knots can be required to control the shape of a curve modification. For instance, inserting knots before and after the parametric position of a point to be translated limits the influence of the corresponding de-Boor points [20], and thus the width of the modified segment (more details in Sect. 4). After $\tilde{\mathbf{T}}$ has been defined, the respective de Boor points $\tilde{\mathbf{D}}$ are adjusted by means of the least-squares progressive iterative approximation algorithm with energy term (ELSPIA) [25], which minimizes the least-square distance to the original curve taking into account a stretching term. The result is a cubic B-spline with single inner knots and de Boor points $\tilde{\mathbf{D}}$, which approximates the original curve sufficiently accurate. Depending on the target shape, the multiplicity of some knots may be increased to three, to produce sharp corners (for specific examples see also Sect. 4).

Local reference frame To achieve visually appealing and consistent modifications, it is helpful to apply the translation of knot points in relation to the intrinsic properties of the respective curve, especially in the context of an automated process. In particular, the curve normal vectors provide information about inward/outward direction: by conversion into DC (see Sect. 3.2) all curves are defined in the way that the normals point inward into the region defined by the respective closed contour, i.e., we have a consistent Frenet-Serret frame in which we apply the translation of the knot points. In our specific implementation, see Sect. 4, we apply translations in normal directions only, but other approaches are possible.

Prevention of intersections and distance transform

Depending on the geometry of the base icon, the translation can lead to self-intersections that have to be prevented, e.g., by locally reducing the length of the translation vector. Our approach uses a conservative intersection prevention scheme, based on Distance Transform (DT) [38] of the base icon. A subpixel precision DT allows to construct a (virtual) skeleton between icon’s curves, which, in turn, is used to constrain the modified curves. More specifically, we restrict transformed knot points to move inside the skeleton’s half-space of the ini-

tial curve point. An implementation for a specific translation along normals is given in Sect. 4.

After curve point translation, applied on an adjusted curve $(\tilde{\mathbf{D}}, \tilde{\mathbf{T}})$, and exploiting the aforementioned functionalities, the resulting new B-Spline is constructed by adjusting the de Boor point positions. Particularly, the corresponding offset vector $\Delta\tilde{\mathbf{D}}$ is computed according to Fowler and Bartels [20]:

$$\Delta\tilde{\mathbf{D}}^T = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\Delta\mathbf{Q}^T, \tag{1}$$

where $\Delta\mathbf{Q}$ is the offset vector to the translated knot points and \mathbf{B} is the B-spline basis matrix.

3.3.2 Color

Controllable color modifications are realized by setting color control points $\tilde{\mathbf{C}}^l, \tilde{\mathbf{C}}^r$, i.e., defining respective locations and color values. Analogously to geometric modifications, the arc-length parametrization is used to facilitate a more intuitive control over color point positions.

Changing color of the original color control points $\mathbf{C}^l, \mathbf{C}^r$ modifies the entire icon subregion, influenced by the respective curve. Inserting new color control points, in turn, creates areas with different coloration. The distance between $\tilde{\mathbf{C}}_j^*$ and $\tilde{\mathbf{C}}_{j+1}^*$ that serve as areas boundaries controls the smoothness of the transition between neighbor colors.

Moreover, the diffusion range of colors on each curve side, that is the area affected by emitted color values, can be limited by insertion of a *diffusion barrier* (see [4]), i.e., a DC that has no own color at least on one side. We create diffusion barriers as DT isolines, where the iso-value limits the impact of the color diffusion (see Sect. 3.3.1).

3.3.3 Shrinking and inflating of icon parts

Our approach allows to modify the size of a specific icon part by shrinking or inflating the respective contour. This type of modification is controlled by offsets to the original curve. Instead of using curve point translations, we define the new curve as isolines with a given offset, analogously to diffusion barriers described above, as this approach is more robust against local curve distortions.

3.4 Post-processing

After a glyph has been constructed by applying the corresponding modifications, the PACEMOD representation is back-converted into a DCI. The de Boor points $\tilde{\mathbf{D}}$ of the modified B-Splines are transformed to Bézier control points using a basis transformation matrix [10]. Accordingly, the parametric positions of the color attributes are remapped onto the intervals of the resulting Bézier splines. For rendering the

DCIs as raster images, we use an extended rendering tool of Jeschke [28] as described above.

4 Periodic modifications

The PACEMOD concept, presented in Sect. 3, performs morphological modifications, i.e., it is basically agnostic to the semantic structure of the input icon. Conceptually, it allows for arbitrary modifications, but providing automated means to achieve meaningful visual effects, e.g., visual variables having metaphoric associations, is not trivial. Here we propose an implementation that comprises a generic set of automated contour modifications, namely the generation of periodic, wave-like patterns.

The suitability of the periodic contour modifications for visual encoding is motivated by the findings of the existing psychophysical studies. Early research on visual perception stresses the role of contours as regions of a high information concentration, especially in the peaks of curvature [2], i.e., wave's peaks. Subsequently, the human discrimination performance related to the contours of observed visual elements has been investigated in several studies, the most of which are limited to regular circular shapes. In particular, it has been demonstrated that such shapes can be distinguished [51] and ordered [11, 26] on the basis of the contour's wave frequency. In a recent study, Presnov and Kolb [42] evaluated the perception of periodic modifications of contour's geometry and color, applied to arbitrary iconic shapes, and deduced an appropriate linear quantization model. Our PACEMOD concept builds on this study, in particular, utilizing the proposed quantization model (see also Sect. 5).

4.1 Periodic geometric modifications

The wave-like modification of the curve geometry is performed by a periodic translation of the knot points along the curve's normals in alternating directions and is defined by frequency, amplitude and waveform (see Fig. 4). In this case, $\hat{\mathbf{T}}$ is composed of knots that are equidistant in terms of arc length, placed at a distance controlled by the given frequency.

The implementation of the intersections prevention mechanism (see Sect. 3.3.1) checks whether a translated knot point $\mathbf{K} + a \cdot \hat{\mathbf{n}}^K$ is closer to another curve point \mathbf{N} than to original knot point \mathbf{K} itself, i.e., if it is located outside its skeleton half-space. In this case, we locally reconstruct a *skeleton point* $\mathbf{S} = \mathbf{K} + a' \cdot \hat{\mathbf{n}}^B$ along the translation vector that is equidistant to the \mathbf{K} and \mathbf{N} , whose amplitude a' is given as the length of a leg in the isosceles triangle with base $\mathbf{N} - \mathbf{K}$, i.e., $a' = \frac{0.5\|\mathbf{N}-\mathbf{K}\|^2}{\langle \hat{\mathbf{n}}^K, (\mathbf{N}-\mathbf{K}) \rangle}$ (see Fig. 3, left). The final amplitude is $a' - \epsilon$, to preserve a free space between the curves. This

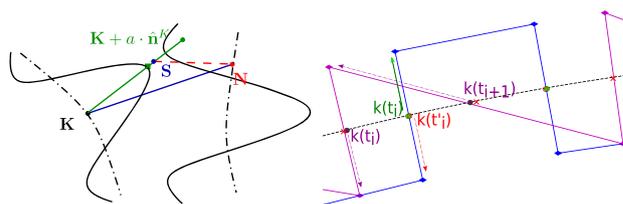


Fig. 3 Left: Prevention of curve intersection. Dashed lines show the original curves and solid lines their sinusoidal modification. The skeleton point \mathbf{S} at the translation vector is equidistant to both curves. The sin peak (green square) has a pixel offset $\epsilon = 3$ from \mathbf{S} . Right: Construction of rectangular and sawtooth shapes of a B-Spline $k(u)$; arrows represent the respective translation vectors. Rectangular: additional knot points with a small offset (red) are added to the initial knot point for the given frequency (green), and are translated in the opposite direction. Sawtooth: the target position for each second knot point is shifted

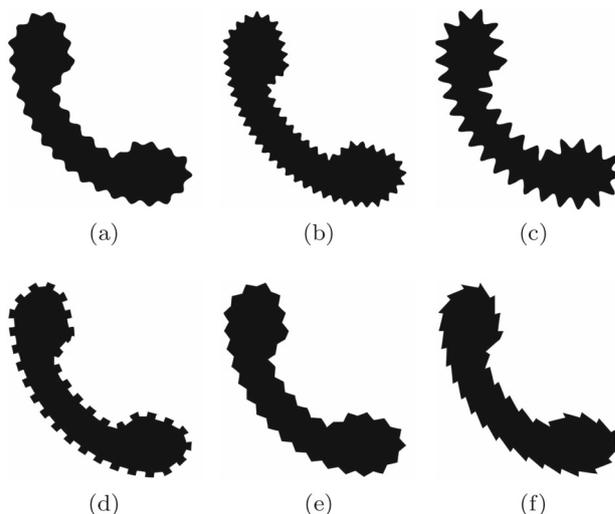


Fig. 4 Examples of periodic geometric modifications. 1st row: varying wave amplitude and frequency; 4a: middle frequency and amplitude, 4c: a higher frequency, 4b: a higher amplitude. 2nd row: different waveforms; 4d: rectangular, 4e: triangular, 4f: sawtooth-like

procedure is applied recursively, as several intersections may occur.

This intersection test is applied to the translated knot points, i.e., the wave peaks, which represent the most protruding parts of the modified curves. While our strategy successfully prevents intersections in almost all cases we observed, extreme cases, e.g., highly curved contours build by multiple curves, which are less suited for PACEMOD, may require an increased skeleton pixel offset ϵ .

Different waveforms can be created by means of additional knots and by shifting the translated positions along the curve (see Figs. 3, right and 4). For instance, in the case of a sinusoidal wave one intermediate knot is inserted between each two 'peak knot points'. These additional knots do not function as curve modification constraints, but restrict the influence of the control points to one period and guarantee a smooth shape (see also 3.3.1). For the triangular,

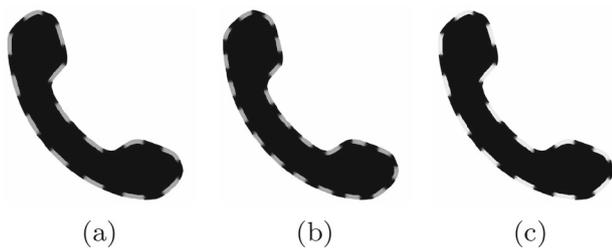


Fig. 5 Periodic color modifications. **5a**: low frequency and dark gray dashes; **5b**: a higher frequency; **5c**: a lighter dash color

sawtooth-like and rectangular waveform, the knot multiplicity is increased to 3 to achieve the C^0 -continuity and thus to produce sharp corners. After a curve modification, each third control point, starting with the first one, is located in a corner, while we place the inner two control points on the lines between the corners to get straight line strips.

4.2 Periodic color modifications

A periodic color modification along the corresponding contour is created by alternating equidistant intervals of two different colors. It is controlled by frequency, i.e., inverse interval's arc length, and respective RGB values. To construct the intervals, we initially compute their virtual boundaries as equidistant arc length positions along the curve, according to the target frequency. This is done analogously to the knot vector calculation for periodic geometric modifications, described above (see Sect. 4.1). Afterward, we place two new color point locations for each interval boundary with a small arc length offset to the left and right side along the curve to get a hard transition between the both colors (see Fig. 2, step 4b and Sect. 3.3.2). For instance, alternating the original curve color with a new one and limiting the color point influence with a diffusion barrier at a small offset (see Sect. 3.3.2), creates the appearance of a dashed line, as shown in Fig. 5.

5 Application examples

In the following we demonstrate the usefulness of our PACEMOD concept and its implementation using periodic, wave-like contour modifications by giving two application examples. First, we demonstrate how the PACEMOD framework allows to apply existing contour-based uncertainty visualization approaches [24, 26] to iconic glyphs, facilitating a more intuitive relation to the application domain. The corresponding examples show visualization of the rain forecast uncertainty (Sect. 5.1). Second, we propose a design of *gradient glyph*, i.e., a glyph that besides main parameters also encodes their changes, e.g., with respect to time. As spe-



Fig. 6 Rain forecast visualization: the uncertainty encoded with sine wave frequency

cific gradient glyph example we choose the visualization of COVID-19 related statistics.

To guarantee the readability and perceptual linearity of the visual encoding, for contour-based visual variables we apply the stimulus-to-perception transformation and the quantization levels estimated by Presnov and Kolb [42]. The size levels have been generated according to Stevens and Guirao [46]. The maps in all examples are created with the Maputnik open source tool.¹

5.1 Uncertainty visualization with iconic glyphs: rain forecast

We visualize two commonly used rain forecast parameters: amount of precipitation and forecast uncertainty or rain probability. The visualized data represent a weather forecast for Europe, September 8th, 2022, collected from WetterOnline² on September 6th, 2022. As glyph's base icon we have chosen a cloud with drops, which is an image often used in forecast websites and apps. To demonstrate the flexibility of our framework, we propose two different design variants.

In the first one (see Fig. 6), the uncertainty is encoded by frequency of a sinusoidal wave, as proposed in [24] and [26]. To generate the corresponding glyphs from the base icon, a periodic geometric modification with fixed amplitude, varying frequency and sinusoidal waveform is applied to DCs that represent a drop contours; see Sect. 4.1. Since the drops in the image symbolize the rainfall, the amount of precipitation

¹ <https://maputnik.github.io/>.

² www.wetteronline.de.



Fig. 7 Rain forecast visualization: the uncertainty encoded with dashed line frequency

can be intuitively encoded with their size. However, the fact that the drops are relatively small glyph's details can impair the size discrimination. To enhance the distinguishability, we couple drop size with saturation and luminance of their color. Technically, the curves representing drops of different sizes are generated from DF isolines (see Sect. 3.3.3) and new colors are created by setting color control points for the inner part of the respective DCs. The mapping of the forecast data to the respective visual variables is represented in Tables 2 and 3.

In the second variant (see Fig. 7), the uncertainty is encoded by dash frequency like in [24]. To create dashed contours of the drops, we apply a periodic color transformation (see Sect. 4.2), alternating intervals of the original drop color and a lighter color. The dash frequency is controlled by interval arc length. To avoid a possible perceptual interference due to the multiple use of color modifications, in this example the amount of precipitation is encoded by number of drops: depending on the parameter value, some drops are hidden by canceling the respective DCs. For the data mapping details see Tables 2 and 3.

Since one of our goals is to enhance the visualization intuitivity by facilitation of metaphoric associations with the related real world phenomenon, we use the natural water color, i.e., blue, as drop color, appropriately selecting the background color. In particular, we selected a grayish green map, which allows for a sufficient contrast with the original dark blue drop color as well as with the light shades of the scaled drops or dash line strokes in the first and second variant, respectively.

Table 2 Visual encoding of amount of rain: by size and color (1st row) and by number (2nd row) of drops

Data	Mapping		
Amount of rain			
	light	moderate	heavy

Table 3 Visual encoding of rain forecast uncertainty: by sine wave frequency (1st row) and dashed line frequency (2nd row)

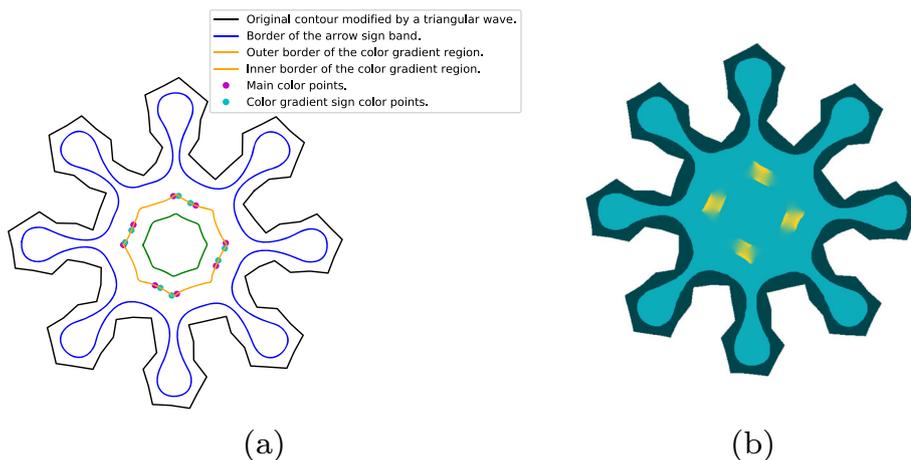
Data	Mapping				
Rain probability, %					
	≥ 90	$[80, 90[$	$[70, 80[$	$[60, 70[$	< 60

5.2 Gradient glyphs

Multivariate data are often given as time series, while a glyph-based visualization usually captures steady data values. Accordingly, for a better understanding of the underlying *dynamics* in the multivariate data, it is also desirable to additionally visualize the temporal changes or trends. The Scientific Visualization methods for representation of time series in a single view, which use additional linear or cyclic axes for the time dimension (see, e.g., [30]), are rather less suitable for a combination with multivariate, glyph-based approaches, since it would lead to a high visual complexity or clutter. There are several visualization approaches, based on abstract glyphs, which visually mimic a function's slope to represent the data changes over time [9, 23, 50]. However, finding an intuitive gradient representation for an icon-based visualization, e.g., exploiting a metaphoric association between a visual variable and the data (cf. Maguire et al. [37]), is a challenging problem, as the main shape is already predefined by the base image.

We address the aforementioned problem, proposing a metaphoric glyph design that allows for visualization of up to two data parameters and integrates a representation of temporal trends in the form of respective derivatives, using periodical contour modifications as additional visual variables. In particular, we utilize the fact that geometric contour modifications are closely related with other geometric visual variables of the glyph, such as size, while color contour modifications are intuitively linked to the glyph's inner color. Furthermore, it has to be assumed that the contour-based visual variables have a weaker pop-up effect. Exploiting these perceptual features, we visually link the primary data to the

Fig. 8 Gradient glyph. **8a**: the glyph structure; **8b**: the resulting glyph after rendering



derived data, i.e., their gradient. In the following, we explain the rules and considerations applied for design of a target glyph from an appropriate icon in detail.

For the sake of brevity, in the following the two dimensional data parameter and the corresponding gradient components are described as $p_i, p'_i, i = 1, 2$, respectively.

1. p_1 is mapped to the glyph's size and p_2 to its inner color, as the visual variables with the highest pop-up effect [7].
2. The goal is to reflect the relation between p_i and p'_i on the visual level, that is the visual variables that encode p'_i need to be intuitively interpretable as *change of size* or *change of color*, respectively.
3. To visualize p'_1 , a periodic, wave-like geometric modification of the glyph's contours with triangular waveform (see Sect. 4.1) is applied. The triangles symbolically represent arrowheads, which indicate the "movement" of the contour, i.e., the glyph's growing or shrinking. The wave amplitude encodes the amount of changes, i.e., the gradient magnitude. The gradient direction, i.e., positive or negative, is visualized by varying the contrast between the triangular contour and the inner part of the glyph, which influences the perception of the "arrow" direction. Particularly, a narrow band along the contour is separated from the rest of the glyph by insertion of a new diffusion curve, which is created from a distance field isoline (see Fig. 8a). In the case of a positive change, the luminance inside of the band decreases, while the saturation increases, leading to a higher contrast in outward direction (see Figs. 8b and 9). In the same way, for negative gradients the band becomes brighter and more desaturated, creating a blur effect and a higher contrast in the outward and inward direction, respectively (see Fig. 9).
4. Analogously, p'_2 is visualized by means of a periodic color modification (see Sect. 4.2). For this purpose, we create a new region in the central part of the glyph, separated by an outer diffusion curve, where a corresponding

color modification is applied, and an inner diffusion barrier (see Fig. 8a). Both are created automatically from the distance field. The diffusion curve is subdivided in alternating segments of the glyph's inner color and a "sign color", which represents the change direction. The length of the sign color segments encodes the gradient magnitude, while its direction is encoded by setting the sign color as one of the both extremes of the color map in use.

Based on these rules, we created a visualization of two of the most conclusive and widely used COVID-19 parameters, 7-days cases incidence and 7-days hospitalization incidence, represented in Fig. 9. The data set is taken by the RKI [45] and refers to the status on June 7, 2022. As the glyph's base icon serves a corona virus icon with the typical spikes. The 7-days cases incidence can be interpreted as a kind of "amount of virus", and thus intuitively encoded by size of glyph. For encoding of hospitalization incidence we use the glyph color with parula colormap, whereby the yellowish hues in the right part, which, correspondingly, represent higher incidence values, are intuitively interpretable as more dangerous. Since in this example the glyph color is used as visual variable, we use a neutral gray background to minimize a possible interference of the color perception. The visual encoding of these data is shown in Tables 4 and 5, respectively. The definition of data intervals is similar to the RKI COVID-19 visualization [44]. Besides the current incidence values, the difference to the previous week is an important parameter, which captures the dynamics of the disease. However, such differences are usually represented as a graph in a separate view, impairing a comprehensive overview of the status and the dynamics of the pandemic. Our glyph design allows for visualization of these differences simultaneously with the current values, by means of contour-based modifications. The corresponding visual encoding is summarized in Table 6. Note that to achieve an uniform mapping, the visual "zero gradient", i.e., no geometric modifications, includes low levels of data change. In

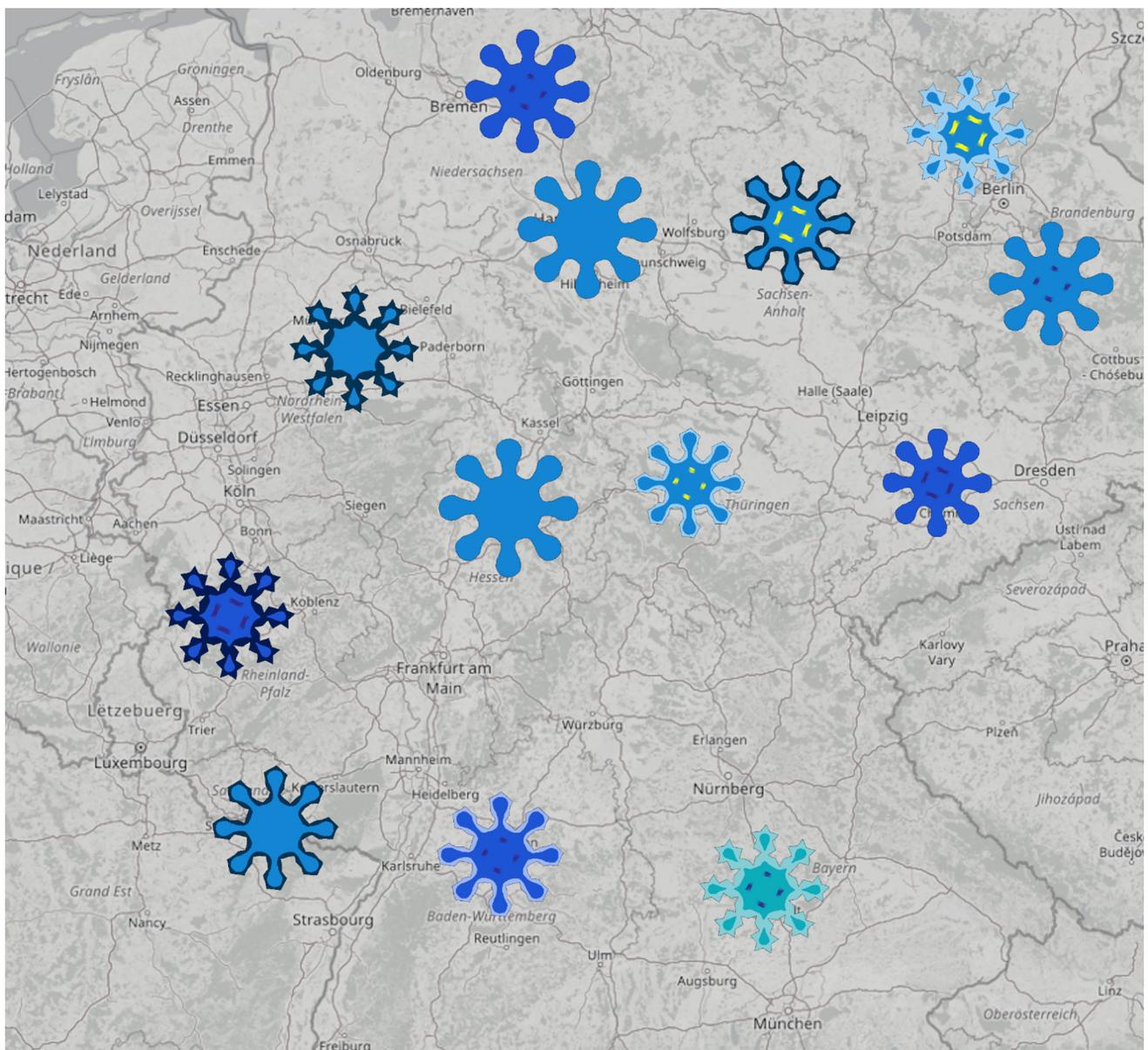


Fig. 9 COVID-19 data visualization: 7-days cases incidence and hospitalization incidence along with the respective changes to previous week. Accordingly to the epidemiological situation as of June 7, 2022, the visualization only uses levels 1, 2 and 3 from Table 4

Table 4 Visual encoding of COVID-19 7-day cases incidence

Data	Mapping					
7 d. cases incid, n./100K popul.						
	< 50	[50, 100[[100, 250]]250, 500]]500, 1000]	> 1000

Table 5 Visual encoding color bar of COVID-19 7-day hospitalization incidence

Data	Mapping					
7 d. hosp. incid, n./100K popul.						
	≤ 1]1, 2]]2, 3]]3, 4]]4, 5]	> 5

total, our approach aggregates in one single visualization the data that are usually spread in four separated views.

To evaluate the usability of the gradient glyph design, we performed an online study with 22 participants, who’s task was to assess the COVID-19 data visualization, given in Fig. 9. This test was mostly distributed among students and university employees. The study was anonymous and no personal data, such as age and gender, have been collected. After a brief explanation of the visual encoding, the participants were confronted with three main blocks of questions. Table 7 provides an overview of the applied questions (in an abbreviated form) and the respective test statistics. More details about test design are provided in the supplementary material.

The first block consists of five question (see Table 7, Q1-Q5), whose primary goal is to evaluate the visualization readability and comprehensibility. Each question presents a state characteristic regarding one or two COVID-19 parameters and a list of five states, for each of them the attendees needed to indicate whether the characteristic applies or not. In the second block (see Table 7, “Trend assessment block”), the goal is to assess the potential of the proposed visualization to facilitate the recognition of trends. In particular, it comprises five statements about statistical trends of the COVID-19 in Germany, and the task is to specify whether they are true or false. While in the first two blocks the map with glyphs was presented along with a legend (analogous to Tables 4, 5 and 6, for more details see supplementary material), the legend was dropped in the third block (see Table 7, Q6-Q8) with the aim to evaluate the learnability, i.e., whether the participants were able to read the visualization without aid after a very learning phase. The questions here are of the same type as in first block. Besides the aforementioned task classification, it has to be taken into account that the question difficulty level varies depending on the number of parameters to assess. In particular, questions Q1-Q3 and Q5-Q6 as well as statements 2 and 5 in the trend assessment block require assessment of one single data parameter, while the remaining questions are two-dimensional.

Table 7 shows the test results as percentage of correct answers per question. In summary, this percentage lies in 42 of 45 cases between 82% and 100%, which demonstrates the suitability of the visualization to convey the encoded information and to serve as basis for trend recognition, despite

Table 6 Visual encoding of COVID-19 data gradient

Data	Mapping				
7 d. cases incid, last week diff., %					
	< -9	[-9, -3[[-3, 3]]3, 9]	> 9
7 d. hosp. incid, last week diff., %					
	< -24	[-24, -8[[-8, 8]]8, 24]	> 24

a very short learning time. In the following, we discuss two cases that are below 75%.

In Q1, answer option e), the percentage of correct answers is only 59%. In this question, the task is to identify the listed states with the lowest 7-day cases incidence, and the correct answers are options d) Thuringia (the glyph close to “Thüringen” in Fig. 9) and e) Saxony (the glyph close to “Sachsen” in Fig. 9). Both glyphs have the same size, encoding the 7-day cases incidence interval [50, 100[. However, the Thuringia-glyph also has a brighter band along its contour, which encodes a negative last week difference for this COVID-19 parameter, which results in a visual “shrinkage”. Due to this visual effect, Thuringia-glyph appears smaller than the Saxony-glyph for 41% participants. In general, the “visual shrinkage” is an intentional effect, aiming at an intuitive encoding of a negative gradient of the main parameter, visualized by size. The critical aspect here is the superposition of the “explicit” geometric scaling and “implicit” visual shrinking that may lead to the misinterpretation as the next smallest size. We will investigate the option to compensate this effect of implicit visual shrinking by applying a slight glyph scaling. Moreover, it has to be taken into account that, in this study, the respective problem appears in the very first question when the participants are still not acquainted with the visualization concept.

The second case of low performance corresponds to Q3, option d), where 73% of the participants gave the correct answer. Here, the task is to select all listed states where the 7-day cases incidence is increasing, which are a) North Rhine-Westphalia (the glyph above “Nordrhein-Westfalen” in Fig. 9) and d) Saarland (the glyph above “Strasbourg” in Fig. 9). The North Rhine-Westphalia-glyph (correctly identified by 86%) has a higher contour wave amplitude, and thus is more salient than the Saarland-glyph, which might have diverted the attention of the participants away from the latter.

To sum up the above discussion, it can be said that the few outliers observed during the evaluation can be explained by interference effects between visual variables. An in-deep investigation of this interference, its dependence on the learning phase duration and its potential compensation by adoptive techniques such as aforementioned glyph scaling requires, however, further special user studies.

Table 7 Evaluation of the COVID-19 data visualization using gradient glyphs

	Ans. a)	Ans. b)	Ans. c)	Ans. d)	Ans. e)
Min. 7-d. cases incid.	NRW: 82%	BR: 95%	ST: 100%	TH: 95%	SN: 59%
Max. 7-d. hosp. incid.	NRW: 100%	BE: 86%	RP: 100%	BW: 100%	BV: 91%
Incr. 7-d. cases incid.	NRW: 86%	TH: 86%	BB: 95%	SL: 73%	BV: 100%
Decr. 7-d. cases AND hosp. incid.	ST: 95%	BB: 86%	RP: 91%	HE: 100%	BW: 82%
Max. decr. 7-d. hosp. incid.	BR: 91%	ST: 100%	BB: 95%	RP: 95%	SN: 95%
Max. incr. 7-d. cases. incid.	BR: 100%	BE: 91%	NRW: 91%	RP: 91%	SL: 91%
Decr. 7-d. cases BUT incr. hosp. incid.	LS: 95%	BE: 91%	TH: 77%	SN: 91%	BV: 95%
Incr. 7-d. cases BUT decr. hosp. incid.	ST: 91%	BE: 100%	RP: 86%	HE: 100%	BV: 95%
The states with the highest 7-d. cases incid. also have the highest 7-d. hosp. incid.					False: 82%
The 7-d. cases incid. tends to increase in the western regions faster than in the eastern ones.					True: 86%
A decreasing 7-d. cases incid. always correlates with a decrease in 7-d. hosp. incid.					False: 91%
A higher 7-d. hosp. incid. always implies a more rapid increase in this parameter.					False: 95%
In South Germany, the 7-d. hosp. incid. is stable or decreasing.					True: 91%

The results represent percentage of the correct answers; bold numbers indicate the answer options where the respective statement does apply. We use the following abbreviations of the German state names: NRW: North Rhine-Westphalia, BR: Bremen, ST: Saxony-Anhalt, TH: Thuringia, SN: Saxony, BE: Berlin, RP: Rhineland-Palatinate, BW: Baden-Württemberg, BV: Bavaria, BB: Brandenburg, SL: Saarland, HE: Hesse, LS: Lower Saxony

6 Conclusion

Summary In this paper we introduced a novel approach for automated contour-based icon modifications. Our approach provides the *PArametric Contour-basEd MODification (PACEMOD)* concept for automated, parametric manipulations of the geometry and color of a given base icon and, thus, contributes to the efficient creation of metaphoric glyphs. In particular, the automation potentially supports user-centered design, allowing a fast or even real-time feedback integration, and thus short iterations in the collaborative visualization developing, e.g., with domain experts. Moreover, we provide a specific implementation of our PACEMOD concept for periodic contour modifications and demonstrate the suitability of the PACEMOD functionalities for transferring existing abstract glyph-based approaches, e.g., in uncertainty visualization, to metaphoric glyphs, as well as for developing of new glyph designs, such as gradient glyphs. In both application examples, the generated glyphs allow for a visually integrated assessment of the multivariate data, which is commonly only achievable using several views. Moreover, our generic approach can be combined with further modifications strategies such as segment-related variation (see the rain drop example). Its potential applications are not limited to the examples, provided in this work, i.e. it can yield benefits when applied for any visualization task that exploits metaphoric associations between data and their representation, especially for visualization of data with an inherent hierarchy that implies different levels of user's attention, i.e. primary/secondary data. A possible example is visualization of patient's parameter by means of anatomical icons.

Limitations On the other hand, the use of icons as base visual element also entails some limitations. First, a more com-

plex shape, comparing with abstract geometric primitives, can affect the distinguishability of visual variables. Second, the necessity to preserve the respective base icon recognizable limits the range of applicable modifications. Therefore, our approach is less suitable for data with a very high number of parameters and for cases where a very high resolution of a single parameter is required, or, at least, it has to be combined with appropriate interaction and data aggregation techniques, e.g. focus+context.

Furthermore, it should be stressed that even if the PACEMOD approach allows for a (semi-)automated generation of a broad design space, the metaphoric linkages on the data parameter level are only guaranteed due to a carefully considered mapping for each specific concept implementation, as has been exemplified with gradient glyphs in Sect. 5.2.

Future work In our future work, we plan to evaluate the corresponding benefits of our approach in specific application domains. Moreover, we see various options to extend our concept, e.g., by developing more sophisticated schemes for locally adaptive applications of shape and color modifications.

Funding Open Access funding enabled and organized by Projekt DEAL. This study was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 262513311 – SFB 1187.

Data and Code Availability The source data for Figs. 6, 7 were derived from www.wetteronline.de on September 6th, 2022. The source data for Fig. 9 are available in https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Daten/Inzidenz-Tabellen.html. There are no further datasets generated and/or analyzed during the current study. The

authors intend to make the glyph generation software publicly available upon acceptance of the paper.

Declaration

Conflict of interest This study was funded by the Deutsche Forschungsgemeinschaft. Besides this, the authors have no conflicts of interest to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Osorio, R.S.A., Brodlie, K.W.: Contouring with uncertainty. In: Proceedings EG UK Theory and Practice of Computer Graphics, pp. 59–66. Eurographics Association (2008)
- Attneave, Fred: Some informational aspects of visual perception. *Psychol. Rev.* **61**(3), 183–193 (1954)
- Bertin, J.: *Semiology of Graphics: Diagrams, Networks, Maps*. The University of Wisconsin Press, Madison (1983)
- Bezerra, H., Eisemann, E., DeCarlo, D., Thollot, J.: Diffusion constraints for vector graphics. In: Proceedings ACM International Symposium Non-Photorealistic Animation and Rendering, pp. 35–42 (2010)
- Boehm, Wolfgang: Inserting new knots into b-spline curves. *Comput. Aid. Des.* **12**(4), 199–201 (1980)
- Bonneau, G.-P., Hege, H.-C., Johnson, C.R., Oliveira, M.M., Potter, K., Rheingans, P., Schultz, T.: Overview and state-of-the-art of uncertainty visualization. In: *Scientific Visualization*, pp. 3–27. Springer (2014)
- Borgo, R., Kehler, J., Chung, D.H.S., Maguire, E., Laramée, R.S., Hauser, H., Ward, M., Chen, M.: Glyph-based visualization: Foundations, design guidelines, techniques and applications. In: *Eurographics (STARs)*, pp. 39–63 (2013)
- Cai, Z., Li, Y.-N., Zheng, X.S., Zhang, K.: Applying feature integration theory to glyph-based information visualization. In: *Proceedings IEEE Pacific Visualization Symposium*, pp. 99–103 (2015)
- Carr, D.B., Olsen, A.R., Pierson, S.M., Courbois, J.P.: Boxplot variations in a spatial context: an omernik ecoregion and weather example. *Stat. Comput. Stat. Graph. Newsl.* **9**, 4–13 (1999)
- Casciola, G., Romani, L.: A Generalized Conversion Matrix between Non-uniform B-spline and Bézier Representations with Applications in CAGD. *Theory and Applications. In: Multivariate Approximation*, Springer (2004)
- Chung, D.H.S., Archambault, D., Borgo, R., Edwards, D.J., Laramée, R.S., Chen, M.: How ordered is it? on the perceptual orderability of visual channels. *Comput. Graph. Forum* **35**(3), 131–140 (2016)
- Chung, D.H.S., Legg, P.A., Parry, M.L., Bown, R., Griffiths, I.W., Laramée, R.S., Chen, M.: Glyph sorting: interactive visualization for multi-dimensional data. *Inf. Vis.* **14**(1), 76–90 (2013)
- Cunha, J.M., Martins, P., Machado, P.: Emojinating: representing concepts using emoji. In: *Proceedings International Conference Case-Based Reasoning (ICCBR) Workshop*, vol. 185 (2018)
- Cunha, J.M., Polisciuc, E., Martins, P., Machado, P.: The many-faced plot: strategy for automatic glyph generation. In: *Proceedings International Conference Information Visualisation*, pp. 71–77. IEEE (2018)
- de Boor, C.: *A Practical Guide to Spline*, vol. 27. Springer, Cham (1978)
- De Oliveira, M.C.F., Levkowitz, H.: From visual data exploration to visual data mining: a survey. *IEEE Trans. Vis. Comput. Graph. (TVCG)* **9**(3), 378–394 (2003)
- Draper, G.M., Livnat, Y., Riesenfeld, R.F.: A survey of radial methods for information visualization. *IEEE Trans. Vis. Comput. Graph.* **15**(5), 759–776 (2009)
- Elder, J.H., Goldberg, R.M.: Image editing in the contour domain. In: *Proceedings IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, pp. 374–381. IEEE (1998)
- Flury, Bernhard, Riedwyl, Hans: Graphical representation of multivariate data by means of asymmetrical faces. *J. Am. Stat. Assoc.* **76**(376), 757–765 (1981)
- Fowler, Barry, Bartels, Richard: Constraint-based curve manipulation. *IEEE Comput. Graph. Appl.* **13**(5), 43–49 (1993)
- Fuchs, Johannes, Isenberg, Petra, Bezerianos, Anastasia, Keim, Daniel: A systematic review of experimental studies on data glyphs. *IEEE Trans. Vis. Comput. Graph. (TVCG)* **23**(7), 1863–1879 (2016)
- Fuchs, J., Jäckle, D., Weiler, N., Schreck, T.: Leaf glyph: visualizing multi-dimensional data with environmental cues. In: *Proceedings International Conference Information Visualization Theory and Applications (IVAPP)*, pp. 195–208 (2015)
- Gerrits, T., Rössl, C., Theisel, H.: Glyphs for space-time Jacobians of time-dependent vector fields. *J. WSCG* **25**, 31–38 (2017)
- Görtler, Jochen, Schulz, Christoph, Weiskopf, Daniel, Deussen, Oliver: Bubble treemaps for uncertainty visualization. *IEEE Trans. Vis. Comput. Graph. (TVCG)* **24**(1), 719–728 (2017)
- He, Shanshan, Daojiang, Ou., Yan, Changya, Lee, Chen-Han.: A chord error conforming tool path B-spline fitting method for NC machining based on energy minimization and LSPIA. *J. Comput. Des. Eng.* **2**(4), 218–232 (2015)
- Holliman, N.S., Coltekin, A., Fernstad, S.J., Simpson, M.D., Wilson, K.J., Woods, A.J.: Visual entropy and the visualization of uncertainty. *arXiv preprint arXiv:1907.12879* (2019)
- Jeschke, Stefan: Generalized diffusion curves: an improved vector representation for smooth-shaded images. *Comput. Graph. Forum* **35**(2), 71–79 (2016)
- Jeschke, S., Cline, D., Wonka, P.: A GPU Laplacian solver for diffusion curves and Poisson image editing. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, pp. 1–8 (2009)
- Jeschke, Stefan, Cline, David, Wonka, Peter: Estimating color and texture parameters for vector graphics. *Comput. Graph. Forum* **30**(2), 523–532 (2011)
- Kehler, Johannes, Hauser, Helwig: Visualization and visual analysis of multifaceted scientific data: a survey. *IEEE Trans. Vis. Comput. Graph.* **19**(3), 495–513 (2012)
- Keim, D.A., Kriegel, H.-P.: Visualization techniques for mining large databases: a comparison. *IEEE Trans. Knowl. Data Eng.* **8**(6), 923–938 (1996)
- Legg, P.A., Chung, D.H.S., Parry, M.L., Jones, M.W., Long, R., Griffiths, I.W., Chen, M.: Matchpad: interactive glyph-based visualization for real-time sports performance analysis. *Comput. Graph. Forum* **31**(3), 1255–1264 (2012)

33. Li, Y.-N., Li, D.-J., Zhang, K.: Metaphoric transfer effect in information visualization using glyphs. In: Proceedings International Symposium Visual Information Communication and Interaction, pp. 121–130 (2015)
34. Liu, Dongyu, Weng, Di., Li, Yuhong, Jie Bao, Yu., Zheng, Huamin Qu, Yingcai, Wu.: Smartadp: visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE Trans. Vis. Comput. Graph. (TVCG)* **23**(1), 1–10 (2016)
35. Shufang, Lu., Ding, Xuefeng, Gao, Fei, Chen, Jiazhou: Shape manipulation of diffusion curves images. *IEEE Access* **8**, 57158–57167 (2020)
36. Lu, S., Jiang, W., Ding, X., Kaplan, C.S., Jin, X., Gao, F., Chen, J.: Depth-aware image vectorization and editing. *Vis. Comput.* **35**(6), 1027–1039 (2019)
37. Maguire, Eamonn, Rocca-Serra, Philippe, Sansone, Susanna-Assunta., Davies, Jim, Chen, Min: Taxonomy-based glyph design with a case study on visualizing workflows of biological experiments. *IEEE Trans. Vis. Comput. Graph.* **18**(12), 2603–2612 (2012)
38. Maurer, C.R., Qi, R., Raghavan, V.: A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* **25**(2), 265–270 (2003)
39. Nocke, T., Schlechtweg, S., Schumann, H.: Icon-based visualization using mosaic metaphors. In: Proceedings International Conference Information Visualisation, pp. 103–109 (2005)
40. Orzan, Alexandrina, Bousseau, Adrien, Winnemöller, Holger, Barla, Pascal, Thollot, Joëlle., Salesin, David: Diffusion curves: a vector representation for smooth-shaded images. *ACM Trans. Graph.* **27**(3), 1–8 (2008)
41. Prassni, Jorg-Stefan., Ropinski, Timo, Hinrichs, Klaus: Uncertainty-aware guided volume segmentation. *IEEE Trans. Vis. Comput. Graph. (TVCG)* **16**(6), 1358–1365 (2010)
42. Presnov, D., Kolb, A.: Perception and quantization model for periodic contour modifications. *J. Imaging* **8**(11), 311 (2022)
43. Risch, J.S.: On the role of metaphor in information visualization. arXiv preprint [arXiv:0809.0884](https://arxiv.org/abs/0809.0884) (2008)
44. Robert Koch Institut. Covid-19-dashboards. <https://experience.arcgis.com/experience/478220a4c454480e823b17327b2bfd4> (2022). Last Accessed Nov. 21st, 2022
45. Robert Koch Institute. Coronavirus sars-cov-2. https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Daten/Inzidenz-Tabellen.html (7/09/2022)
46. Stevens, S.S., Guirao, M.: Subjective scaling of length and area and the matching of length to loudness and brightness. *J. Exp. Psychol.* **66**(2), 177 (1963)
47. Sun, Timothy: Thamjaroenporn, Papoj, Zheng, Changxi: Fast multipole representation of diffusion curves and points. *ACM Trans. Graph.* **33**(4), 1–53 (2014)
48. Surtola, H.: The effect of data-relatedness in interactive glyphs. In: Proceedings International Conference Information Visualisation, pp. 869–876. IEEE (2005)
49. Ward, M.O.: Multivariate data glyphs: principles and practice. In: *Handbook of Data Visualization*, pp. 179–198. Springer (2008)
50. Wickham, Hadley, Hofmann, Heike, Wickham, Charlotte, Cook, Dianne: Glyph-maps for visually exploring temporal patterns in climate data and models. *Environmetrics* **23**(5), 382–393 (2012)
51. Wilkinson, F., Wilson, H.R., Habak, C.: Detection and recognition of radial frequency patterns. *Vision. Res.* **38**(22), 3555–3568 (1998)
52. Xie, Guofu, Sun, Xin, Tong, Xin, Nowrouzezahrai, Derek: Hierarchical diffusion curves for accurate automatic image vectorization. *ACM Trans. Graph.* **33**(6), 1–11 (2014)
53. Ying, L., Tang, T., Luo, Y., Shen, L., Xie, X., Lingyun, Y., Yingcai, W.: Glyphcreator: towards example-based automatic generation of circular glyphs. *IEEE Trans. Vis. Comput. Graph. (TVCG)* (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Dmitri Presnov received a Master of Science degree in computer science from the University of Siegen, Germany, in 2017. He is currently working as a researcher with the Computer Graphics and Multimedia Systems Group, University of Siegen. His research interests are 3D scene reconstruction on mobile environments and medical visualization.



Marlena Berels is studying computer science with a specialization in visual computing at the University of Siegen, Germany. She is currently writing her bachelor's thesis in the field of visualization.



Andreas Kolb received the PhD degree from the University of Erlangen, Germany, in 1995. He is currently the Head of the Computer Graphics and Multimedia Systems Group, University of Siegen, Germany. His research interests include computer graphics and computer vision, including real-time simulation, processing, accumulation, and visualization of sensor, flow and medical data.