

— Metamorphers —

A Formalization of Transitions in Illustrative Molecular Visualization

A, B, and C

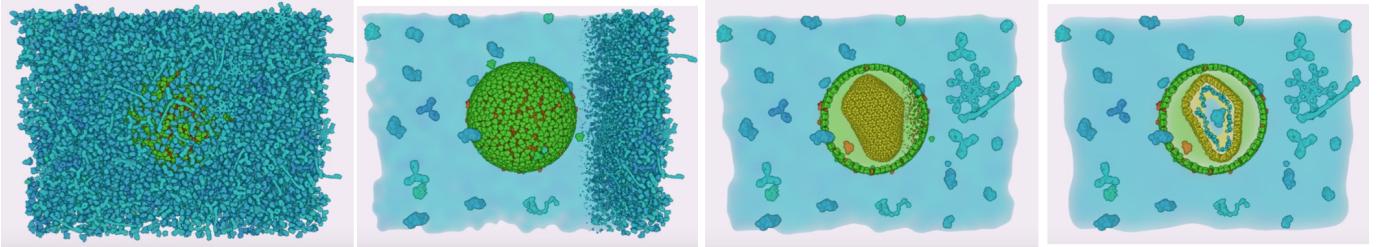


Fig. 1: Metamorphosis to a schematic representation of an HIV particle in blood serum.

Abstract— Illustrators in molecular biology often use animated transitions as a means to communicate complex phenomena. While these animations are popular, they still have to be created manually in 3D modeling software. We therefore provide illustrators with a more streamlined way for creating such transitions between various representations of molecular data.

We propose a formalization of the process of specifying transitions to arbitrary representations in the form of a conceptual pipeline. Data representations are specified in such a way that they form a continuous space. The first three stages of the pipeline are responsible for creating this continuum. The final four stages of the pipeline are responsible for presenting the continuum. This continuum can be arbitrarily sampled in order to create a transition between two representations, e.g., densely, in the form of animation, or sparsely, in the form of small multiples.

We demonstrate our concept on exemplary use cases by creating multiple operators, so called metamorphers, for each pipeline stage. These metamorphers can be combined to achieve different types of transitions for arbitrary molecular data sets. We present three transitions that we designed in collaboration with illustrators, based on the data set of HIV: the explosion of inner and outer cellular compartments, a novel illustrative rendering approach for automatic schematization of mesoscopic data, and a transition to an infographic view, conveying quantitative relationships. In a qualitative discussion of our results, we received positive feedback from domain experts in illustration.

Index Terms—Molecular Visualization, Illustrative Rendering, Animation.

1 INTRODUCTION

The traditional approach for the visual communication of scientific insights is handled by illustrators and animators. Their goal is to convey complex phenomena in an accessible way, and to attract the interest of broad audiences. In recent years, we have seen a rapid increase in the communication of topics from molecular biology [cite[drew berry, gj, david bolinsky, Gal McGill, janett iwasa]], such as the splitting of DNA, the transport of oxygen in the blood stream, or the comparison of molecular structures within a cell. Such animations are used in teaching undergraduate level biology worldwide.

When illustrators want to convey, for instance, the inner and outer structures of a cell, they face the challenge that the structures of their data are very densely packed. The dense data makes it very difficult to convey inner and outer structures at the same time. Therefore they employ means for occlusion handling, such as exploded views, in order to make all important aspects of the data visible. Illustrators often want to convey the entire variety of different molecules within a cell in form of an atlas-based view [cite some atlas]. However, a cell contains in reality many thousand instances of the same molecular structures. To achieve such an atlas based view, an illustrator does not want to display every single molecule but would rather draw various structures at

sufficiently large scale so that the viewer can understand the structural details. In these examples, the illustrator represents abstracted spatial information within same spatial frame of reference. We denote this as a transformation *within* the same visualization space (Figure 2a). In other cases, illustrators want to convey quantitative instead of spatial information, e.g., how many different molecules of each type a biological entity contains, or how a functional relationship between these molecules would look like. In the former case, a bar chart could be used as representation of the quantities. A node-link diagram could be chosen in the latter case to convey the relationships. In such cases, the illustrator would abstract the biological structure from a spatial representation to a quantitative or relational one, which we denote as a transformation *across* visualization spaces (Figure 2b).

For biological entities, a "canonical" representation, especially for non-experts, is the one closest to their mental model that is formed by images from photography, microscopy, and XXX(could not read). The mental model is therefore anchored within the same visualization space as the actual biological entity. If no obvious relation between a mental model and a particular representation exists, the viewer needs guidance on how to relate their mental model to this transformed representation, in order to comprehend the other (transformed) representation form.

In scenarios where interaction is possible or desired, e.g., in an interactive analysis scenario, approaches such as coordinated multiple view (CMV) visualizations can be used for conveying the relations between two representations, e.g., via brushing and linking.

In cases, where interaction is not feasible, such as in presentation scenarios, this guidance is often realized by depicting a transition from a representation A to a representation B, e.g., in the form of static se-

- Author A is with Institute X. E-mail: a@x.com.
- Author B is with Institute Y. E-mail: b@Y.com.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.

Digital Object Identifier: [xx.xxxx/TVCG.201x.xxxxxxx/](https://doi.org/10.1109/TVCG.201x.xxxxxxx)

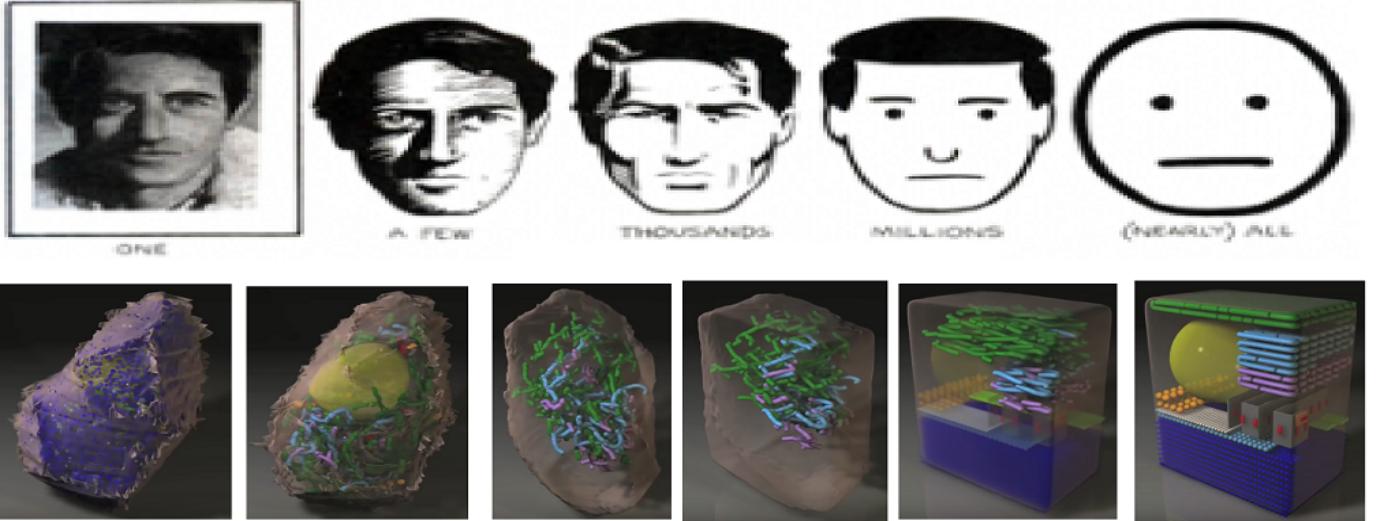


Fig. 2: Examples of transitions within the same visualization space (a), across different visualization spaces (b).

quence of snapshots/small multiples, or an animation/animated transition. We denote such a transition as a metamorphosis and the individual building blocks of the metamorphosis as metamorphers.

Animations are a powerful tool for conveying these metamorphoses, since they are capable of displaying such a transition between two representations in a continuous way. The continuous presentation supports an intuitive understanding of the relationship between representation states. The presentation is both self-explanatory, and enjoyable to audiences [13]. Despite their immense popularity in the visual communication ... (cannot read), animations are also critically regarded by the visualization community. Tversky describes the fleetingness of the displayed information and the potential sensory overload and distracting nature of badly designed animations.

Alternative solutions for conveying transitions alleviate the fleetingness of the displayed information by sacrificing the continuous aspect of the representation. Examples are narrative sequences of small multiples or static images that supplement the missing information of the continuity with glyphs. While solving the problem of fleetingness, they have to be well designed to be understandable.

Once a transition from one representation to another one is planned/conceptualized, there is little aid in modelling tools to realize it. Animators employ key frame animation and create these keyframes manually which is a laborious and time-consuming task. Considering that new discoveries are made frequently in molecular biology, these representations and their transitions have to be re-modeled when new insights are found, which makes the effort for maintaining them even higher.

We therefore set our goal to assist illustrators by eliminating the need to manually author their target representations and the transitions to them. We supply illustrators with a pipeline for creating continuous target representation as well presenting the transitions to them.

Our primary contribution is a formal description of a uniform method for creating continuous target representations for spatial biological data sets, as well as presenting the transitions to these representations - describing the complete metamorphosis of the data representation, so to say.

Data representations are specified in such a way that they form a continuous space. The first three stages of the pipeline are responsible for creating this continuum. The final four stages of the pipeline are responsible for presenting the continuum. This continuum can be arbitrarily sampled in order to create a transition between two representations, e.g., densely, in the form of animation, or sparsely, in the form of small multiples. Each stage supports operators, so called metamorphers, that are responsible for one aspect of the metamorphosis.

As our secondary contribution, we demonstrate a proof-of-concept

implementation of this pipeline based on three use cases that we developed in collaboration with illustrators. One of these results, a novel illustrative rendering approach for automatic schematization of mesoscopic data, is our third and final contribution.

2 RELATED WORK

Animation and transitions between visual and data representations have been employed in many contexts. They serve as powerful tools for the dissemination of complex relations in space, time and abstract dimensions. They are frequently used in visual story telling and for the depiction of correspondences between data representations as well as visual representations.

2.1 Visual story telling

Kosara and Mackinlay [9] note that for a long time the focus of visualization research was on exploration and analysis but that presentation of findings especially using elements of story telling should be a research focus of equal importance. They list prominent examples of story telling that include animated transitions for trend analysis and dissemination.

Robertson et al. [13] have looked at trend visualizations in depth. Three methods including animated transitions were analyzed. They found that animated transitions of data was reported to be more enjoyable and exciting to users. They also found that it was significantly faster when used for presentation but less exact and less effective for analyzing data. These results encouraged us to use animated transitions to facilitate presentation of complex molecular data.

Segel and Heer [14] analyze story telling with narrative visualization and point out different approaches. They distinguish between author- and reader-driven elements in a narrative and speculate that "exploration of transitions between them presents an exciting area for researchers and practitioners". Along these lines Wohlfart and Hauser [17] presented a guided interactive volume visualization approach. Their system enables authoring and editing of visualization stories that give the user partial control over the exploration.

Ma et al. [12] describe the need of new scientific visualization methods that can be used for scientific story telling. Among other issues they suggest that novel transitions between different coordinate systems are needed to address the overwhelming complexity involved in today's scientific story telling.

2.2 Transitions between visual representations

Kosara et al. [10] use 3D scatter plot matrices to establish a link between physical layout and the abstract dimensions of the data. Their

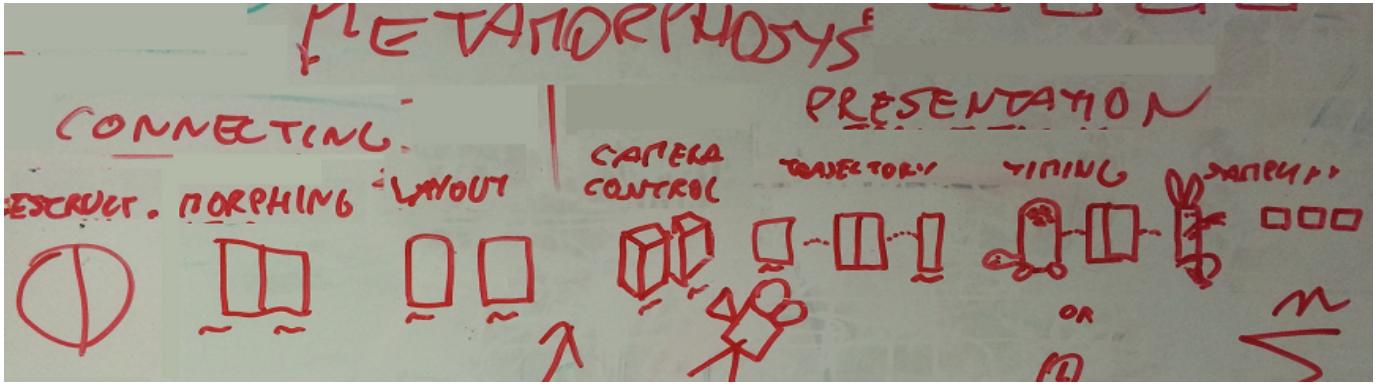


Fig. 3: The metamorphosis pipeline. Continuum creation: data restructuring, morphing, layouting. Continuum presentation: camera control, trajectory, timing, sampling.

work is an early attempt to connect information visualization and scientific visualization using points as data primitives. Elmquist et al. [3] present an exploration technique for multi-dimensional data. They place scatter plots on the sides of a dice and animate the transitions when the dice is rotated.

Guilmaine et al. [4] compare different animated transitions of tree structures. They find that hierarchical animation is better suited for tracking of changes.

Basch [1] describes possibilities for animated transitions between volumetric rendering and abstract views like histograms and scatter plots. They use staggered animation to reduce occlusion. Hurter et al. [7] present a more general technique that interpolates data points between different views which are projections of the original data dimensions. They demonstrate applications for volume data as well as for images. Their approach addresses occlusion with interaction methods like brushing and locking of data points. Heer and Robertson [5] show animated transitions between different visual representations of statistical data. Their work makes extensive use of staging to reduce occlusion and clutter. Our approach is based on several findings of these previous works and formalizes animated transitions between spatial and abstract attributes.

Huron et al. [6] present the visual sedimentation metaphor for the animation of data streams. This metaphor allows to transition from discrete visual elements to a continuous representation. We employ a similar metaphor for the transition of spatial objects to abstract charts.

Exploded views [2, 11, 8] have been used in different contexts as a technique for illustrating spatial or hierarchical relations of parts of an object. They are frequently used in animations or interactive systems and reported to be effective in the dissemination of complex layouts. Our approach is similar to exploded views, since it animates the decomposition of the object into its parts. However, the spatial displacement is not only driven by the visibility of the individual parts, but also by additional data attributes.

2.3 Alternatives to animated transitions

Small multiples [15], coordinated multiple views, and static images in conjunction with annotations, traces and glyphs are alternatives that are typically used to visualize transitions, trends, correspondences or sequences [13]. Tversky et al. [16] summarize cognitive studies on the benefits of animation. Although, they conclude that animation alone has not been convincingly demonstrated to be superior to static illustrations, other findings of their study suggest a direction for research on animations in visualization. They report that animation together with basic interaction methods like pausing, partial re-playing, zooming and change of perspective might be the key to enhance the effectiveness of animation. Further, they acknowledge that the alternatives to animation (such as sequences of static illustration) are surprisingly hard to design and are therefore not an easy target for computer automation.

3 METAMORPHOSIS FORMALIZATION

The context of our research is an explanatory visualization of large-scale structural models of viruses and bacteria that conveys relationships between various abstracted or transformed visual representations of the models without any need for user interaction. Such visualization techniques can then be easily employed by scientific illustrators and animators for easily accessible and engaging content authoring for popularising and conveying science to non-expert audiences.

Each illustration which communicates a relationship starts with the two representations of the model. The first, source representation, is typically close to the mental model of the audience. The second, target representation illustrates different aspects of the presented model, but its relationship to the source representation might not be obvious. In that case, the illustrator needs to manually create the visual transition between the two representations. We present a method for creating continuum between two arbitrary model representations, or metamorphosis, and using it to illustrate the relationship between these representations. The metamorphosis is created automatically through describing the target representation, thus alleviating the illustrator from creating it manually.

Figure 4a shows the source and the target representations. They both communicate certain aspects of the model, but they are disconnected, their relationship is not immediately clear. Our pipeline consists two steps. In the first step, we create a continuous space between these representations. However, this low-level connection is not yet ready for illustrating the relationship between the representations (Figure 4b). Therefore, the second step of the pipeline prepares this continuum for the presentation to the intended audience (Figure 4c). In this step, the metamorphosis is finalized so that it illustrates the relationship between the source and the target representations.

The metamorphosis is carried out by applying a sequence of operators, which we refer to as *metamorphors*.

3.1 Continuum Creation

In the first step of the pipeline, the continuum between two model representations is created. In this section, we describe the three stages of the continuum creation, with examples of metamorphors which can be used in these stages.

3.1.1 Data restructuring

Biological data models are organized in a hierarchy, which corresponds with the semantics of abstraction of the modelled phenomena. This hierarchy often differ in both representations. Therefore, the first step of connecting the representations is to match their semantic hierarchy. In this step of the pipeline, the illustrator defines data subsets within the source model representation, which match the semantics of the target representation, thus restructuring the data. Additionally, some of the data subsets can be duplicated, if the target representation requires it. All the data subsets are then organized in a hierarchy with a tree structure, or a scene graph. All the subsequent pipeline stages can operate on any node of the scene graph, giving the illustrators the flexibility they are used to from the 3D modelling software.

Examples of metamorphors used in this stages are implicitly defined cutaway objects, such as those proposed by Le Muzic et al. ??, which specify spatial data subsets, or various data clustering algorithms.

3.1.2 Morphing

An important aspect that might obfuscate the relationship between different representations of the biological data are the shapes, or visual representations, of the individual data elements. For instance, molecules in the mesoscale biological models can be displayed as space-filling models, stick models, or one of many different representations commonly used in molecular graphics. Alternatively, molecules can be represented by abstract shapes, such as balls or cubes illustrating their volume.

In this stage of the pipeline, the relationships between the shapes of the corresponding data elements are specified. For this purpose, two types of metamorphors can be applied.

Object-space morphing metamorphors are used to transition between 3D shapes of the visual representations of the data elements. For instance, in the molecular data, this is achieved by continuously repositioning the individual atoms of a molecule to form the shape of a different molecule. If the atom counts of the two molecules differ, multiple atoms are placed at the same spatial location to match these counts before the object-space morphing is applied.

Image-space morphing metamorphors transforms the visual representations of the data elements in the image space, for instance through alpha blending or other image-processing operations. These metamorphors are used when the two representations cannot be adequately matched in the object space.

3.1.3 Layout

Depending on the intended message, the illustrators often need to reposition data elements within the visualization. This might be done as means for occlusion handling, e.g., by exploding dense data to reveal the internal structures, or to simplify noisy or complex spatial arrangement of the data elements to better fit the viewer's mental model. Elements could be positioned so that they represent an altogether different visualization space, such as a network that shows interactions between molecules.

To allow the illustrators to create transitions between different data representations, the spatial arrangements of the data elements in both representations need to be matched. This is achieved by transforming the positions, rotations, and scaling of the data elements through *layout* metamorphors. For instance, such metamorphors can arrange the data elements into various 3D volumes, such as cubes or spheres in order to illustrate their relative counts, or align them side by side for comparison.

The layout metamorphors operate on the individual nodes of the scene graph down to the individual data elements (e.g., molecules), so that whole data subsets can be transformed at once. Additionally, the layout metamorphors can be arbitrarily stacked to achieve wide range of spatial transformations.

3.2 Continuum Presentation

After the continuum between the two data representations is created, it needs to be processed so that it adequately communicates the intended message. This processing takes place within the second step of the pipeline, consisting of four stages.

3.2.1 Trajectory

While the layout stage of the pipeline defines the relationship between data elements in different data representations, the actual transition between them is not yet specified. It is necessary to define the trajectory along which the data elements move in order to transit from the source representation to the target one. This is done in the first stage of the continuum presentation step.

The trajectories are defined as arbitrary sets of control points between the source and target transformations of the data elements. These control points can form various curves along which the data elements move during the metamorphosis. The continuity of the spatial transformations is achieved through linear interpolation of the translations and scaling, and spherical linear interpolation of the rotations.

The metamorphors of the trajectory stage can be used to structure the transition, such as through edge-various bundling methods. For instance, the molecules of the same type can move to their target positions along similar paths in order to minimize the visual clutter created by the transition.

3.2.2 Timing

In order to present the transitions specified in the morphing and the layout stages to the viewers, illustrators need to create appropriate temporal arrangement for them. Such arrangement encodes the information about the chronology of the illustrated events. Additionally, illustrators use the speed at which individual transitions occur as means to suggest their importance for the communicated message, as well as to draw viewers' attention to different parts of the illustration.

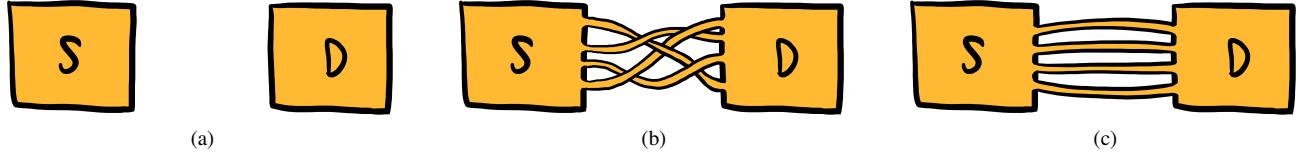


Fig. 4: The continuum between two representations describes the metamorphosis from one representation state to another: a) a source and target representation. b) the continuum creation connects both representations. c) the continuum is restructured for presentation.

The temporal arrangement is designed through metamorphers of the timing stage. They operate on so called *time curves* associated with each node in the scene graph. These curves are used as transformations of the data elements' positions within their specified trajectories as functions of time.

The metamorphers are used to specify the starting time of each transformation, as well as its speed. As such, they can be used to create various temporal effects, e.g., ease-in or ease-out curves for the movement of the data elements or entire data subsets. The time curves can be also modified to make some of the elements stop in the middle of the specified trajectory, or to reverse their movement.

3.2.3 Camera Control

3.2.4 Sampling

4 USE CASES

Real world illustrators now can use this pipeline of metamorphers to create continuous representations that depict a desired metamorphosis of a given source representation to an intended target representation by combining the metamorphers on each pipeline stage that create the intended relations between the source and the target representation. To actually enable an illustrator to apply our method, we created a proof of concept implementation of our pipeline. In order to apply our method to solve relevant problems, we approached an illustrator and, in collaboration, created three different use cases for a data set of HIV. This data set represents an HIV particle contained within blood serum (see Figure XXX). The three questions that we helped to answer with our demonstration are: 1) what does the three dimensional structure of the virus and its inner compartments actually look like and how do they relate to each other hierarchically? 2) what types of molecules are contained in which compartment of virus, how do they look like, and in which approximate quantity are they present? 3) how large is the volume of each compartment and the molecules contained within in respect to each other?

4.1 Explosion of Molecular Structures

Since molecular structures of the HIV data are very densely packed, it is impossible to inspect the outer and inner structures of the virus at the same time. Since the four structures that we want to show, hierarchically contain each other, much like a Russian matryoshka doll, we chose to transform the virus representation to an exploded view (Figure ??), to answer our first question.

We can achieve the transition of the HIV data to an exploded view in the following way: First, we want to create the relation of the original data to the four compartment structures of the target representation. In the data re-structuring stage, we therefore need to create four subsets from our original data that correspond to these compartments. Since each compartment contains molecules of differing types, we implemented the *type range* metamorpher that creates subsets that contain all given molecules that match a specified type range. These four subsets are added to the root node of our data that contains all molecules. Since three of the four compartments that we just created are exploded in our desired target representation, we further have to spatially subdivide these compartments along a plane. We therefore implemented the *plane splitting* metamorpher that checks the position of a given list of molecules in respect to a plane that the user can freely position in the scene. Molecules are then assigned to two new subsets, depending on which side of the plane they are situated. The scene hierarchy is

therefore extended by two additional child nodes containing the spatial subdivision for each subset that we just spatially subdivided.

Since the visual representation of individual molecules does not change across the source and the target representation, we do not need to apply any metamorphers in the morphing stage of our pipeline.

In order to relate the original unexploded data to the final exploded representation, we need to call *translation* and *rotation* metamorphers in the layouting stage that create the translation (and slight rotation) of the explosion for each pair of spatially split subsets (Figure ??b). An additional translation moves each inner compartment to the left of its previously encapsulating compartment (Figure ??c).

We now created the continuum between the original unexploded and the exploded target representation. In order to present the transition in a pleasant way, we follow the remaining steps of our pipeline.

Since both, source and target representation, remain within the same view frustum, and we do not want to explicitly point the camera on any specific details, we apply no metamorpher in the camera control stage.

Our original data differs from the exploded target representation only in position and topology (the split compartments). To present the transition, we therefore want to move all compartment subsets as a unit - they should keep their shape during the transition. In the trajectory stage, we therefore apply the *linear interpolation* metamorpher to achieve a linear transition along the continuum between source and target representation.

In the layouting stage, we apply a total of XXX layout metamorphers/operations to create the relation to the exploded representation. If all these operations would be executed at the same time, the transition could be hard to follow [cite Heer paper here]. In the timing stage, we can make the presentation of the transition more pleasant, by applying a *delay* metamorpher. The delay metamorpher adjusts the time curve of each compartment subset so that each subsequent explosion starts only after the previous is finished. To suggest the opening motion of each exploding compartment in the final representation, we apply a *[need fitting name]* metamorpher that adjusts the time curve of a few random molecules in such a way, that they do not finish their transition. During the transition, the opening compartments thus leave a trail of molecule "breadcrumbs" behind.

The only thing that is left to do now is to sample the continuum representation that we created. We chose a *dense/continuous sampling* metamorpher to create a ten second animation with twenty four frames per second for the supplemental video. For the Figure ?? we applied a *sparse sampling* metamorpher [or we took stills from this video?][or we have one sampling metamorpher that we used for video and figures but with different parametrizations - how to spin/sell this?].

4.2 Schematization of Molecular Structures

The data consists of 42 types of molecules, that have a total of XXX number of instances, consisting of a total number of YYY atoms.

4.3 Representation of Quantitative Relations

[this is the old usecase section]

4.4 Schematization of Molecular Structures

In scientific illustrations, atlas based views schematic representations of cellular structures are used to give an insight of the molecular and

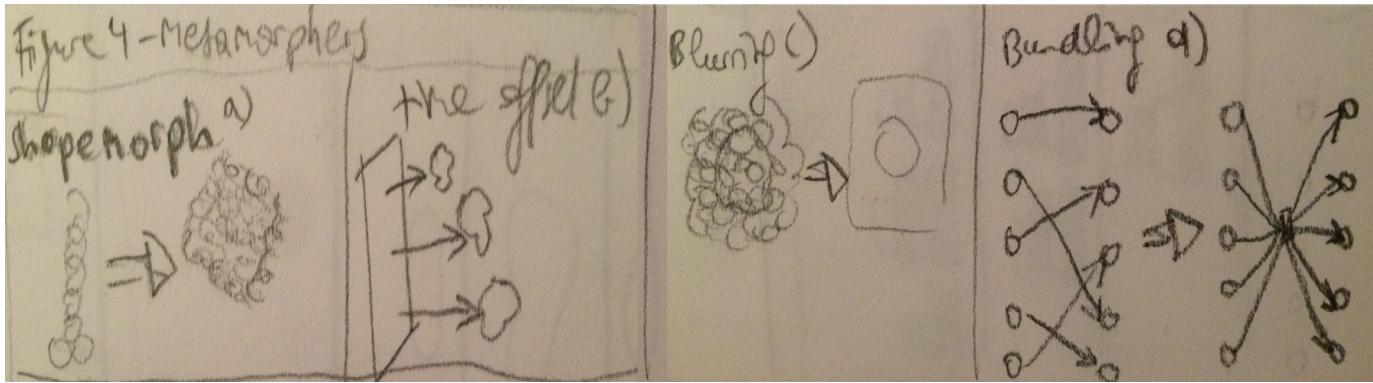


Fig. 5: Examples of different metamorphers: a) shape morphing, b) blurring, c) time offset, d) some layout

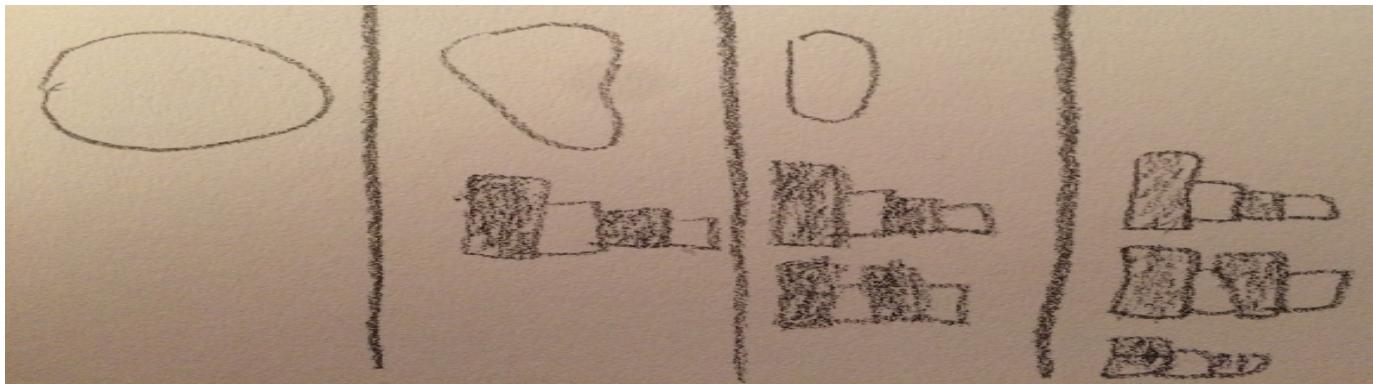


Fig. 6: Infovis

atomic composition of these cells. The focus lies therefore on displaying individual molecules to convey their atomic composition, as well as on selected constellations of molecules to convey the structures that they collectively create. However, the original data model consists of millions of mostly noisy distributed atoms. To achieve this schematic representation therefore, one the one side, specific molecules have to be chosen as representatives of their type to be displayed at a sufficiently large scale (focus), while suppressing the noisy distribution of millions of atoms (context). On the other side, molecules that do not have a noisy distribution but form coherent structures, such as cell membranes, have to be preserved. This representation sacrifices the low level detail of the noisy atomic distribution, i.e., the exact count, position and shape of individual molecules. It aims to preserve the shape and size of compartments as well as molecule type overview, compartment relation (Zugehörigkeit), and molecule shape. This metamorphosis corresponds with a transformation within the same visualization space.

@discussion of results: transition = more intuitive to follow.

because this type of automatic illustrative schematization has not been presented/solved yet in literature, we present it as our tertiary contribution...

4.5 Representation of Quantitative Relations

For the explanation of the molecular data, often also the relation of quantitative aspects is important for its understanding, e.g., how many different molecules of each type does the HIV particle contain, or how much of the volume within the particle does each type occupy? Graham Johnson answered this question in his award-winning video demonstration of the decomposition of an XXX cell. In order to achieve this sort of representation, the molecular data has to be transformed into a different visualization space. A stronger abstraction than in the previous two examples takes place. To convey these quantitative information about aggregated volumetric and count information,

shapes and positions of compartments and molecules are sacrificed. Since this transformation is across visualization spaces, the target representation is harder to relate to the original representation.

@discussion: here, large distance between representation forms in visualization space - a continuous transition is therefore especially suitable for conveying the transition..

4.5.1 Grouping by Type - use case C

We need to visually or spatially transform all molecules of the same type in a uniform fashion, e.g., to spatially transform them into a collective shape. [refer to all usecases here?] The output of this metamorpher is therefore a list of subsets that contain entities of the same type. The metamorpher parses a given list of entities and creates a new subset for each new type-ID that it finds. Entities of the matching type are then assigned as children to this subset.

4.6 Schematization of Molecular Structures

4.6.1 Applied Metamorphers

-data transform: 1) [type range grouper] create three high lvl subsets: the blood serum, the membrane, the capsid 2) [copy] the entire partitioned data set because we will render it in two different representations => results in copy A & B

-annotations: do the circular annotations?

-visual transform: A: blurrin & increasing contrast of alpha channel for nice sharp edges of each subset individually B: scalin

-layouting (spatial transform): A: renderin them on top of each other (occlusion management) B: rotatin

-navigation: not necessary, because we stay in the same visualization space / spatial extents

-trajectory: B: linear interpolation between rotations

-timing: staged per compartment distance field left to right for each

-sampling: continuous

4.7 Representation of Volumetric Relations

4.7.1 Applied Metamorphers

—data transform: 1) [type range grouper] create 3 hierarchical compartments by type range 2) [single type grouper] split all three by type = \subseteq subsets containing all molecules of same type for each hierarchical compartment

—annotations: 3) [txt-label] for each type subset 10) [representative molecule for type subset]

—blending: 4) [slice-flatterer] create target rep for each molecule with blending: slice with height representing the molecule volume

—layouting: 5) [bar layout] for each type compartment - the layout puts all molecule slices on top of each other to create histogram bars
6) the [line layout] puts all the bars within a compartment side by side
7) the [translate layout] then moves the new center for the next line layout for the next compartment

8) layout: put label below each bar 9) layout: put rep below label

—trajectory: 10) [top-down-fall-in-component]: for each molecule in bar subset, we add a control point to the top of the bar, so that the molecule seems like falling into the bar instead of going straight from its original position to the designated position within the bar.

—timing: 11) [distance offset] a plane can be positioned in space. the distance of each molecule/compartment to this plane is measured and used for a the creation of a time offset that specified when each molecule is allowed to start its transition

12) timing: make label & rep appear when animation is finished

—guided navi: 13) [whole-data-in-view-keeper] update the camera pos & lookat to keep all molecules in the view frustum.

—sampling: 14) continuous sampling

4.8 Metamorphers

[just for reference] [describes PoC components that we implemented per pipeline stage]

[should we describe beforehand what (which representations) we try to achieve with these implementations?]

4.8.1 Partitioning

spatial subselection by clipping plane/object,

splitting into types,

splitting into type ranges

cloning

—not showcased but implemented: *spatial splitting operations into equal parts (pie splitter),

4.8.2 Annotations

text labels, icons (textures), maybe: data representatives (for each type), optional addon to each operator: connectors between annotation & assigned subset

4.8.3 Blending / Morphing

blur into collective shape, morph,

—not showcased but implemented: blending to texture

—other not implemented & not used examples: unfolding

4.8.4 Layouting

bar, line, translate, rotate, scale, schematization layout = combi of scale & rotate but selection of elements where what op is applied is very specific

—not showcased but implemented: slice, sphere, circle

4.8.5 Camera / Guided Navi

[whole-data-in-view-keeper] update the camera pos & lookat to keep all molecules in the view frustum.

other examples:

4.8.6 Trajectory

linear: from A straight to B, bundling/transit points

—other not implemented & not used examples: curved

4.8.7 Timing

offset function: delay timing by distance to plane & staging by type ... anything else?

—other not implemented & not used examples: offset according to: inherent spatial parameters (position, distance to other entities, density)

4.8.8 Sampling

continuous sampling,

—other not implemented & not used examples: sparse (comic strip / narrative sequence),
post-processing operators: motion blur,

5 IMPLEMENTATION

6 DISCUSSION

6.1 Discussion of Results

6.2 Expert Feedback

6.3 General Applicability of Our Approach

compatible data formats: point cloud, volume, polygonal?

compatible data types: dense hierarchical data with lots of repeating entities of a few dozen types

6.4 general discussion

6.4.1 additional outcomes

side products of our pipeline - free of charge / easily achievable with minimal additional effort: sequence: comic strips, static image with glyph

6.4.2 animation

tversky: congruence principle: the structure and content of an external representation should match the structure and content of the desired mental representation.

apprehension principle: the structure and content of the external representation should be readily and accurately perceived and comprehended.

6.4.3 things that should be preserved:

depending on the usecase / representation - not all properties are essential for each representation form.

- instance level: position of instances
- type level: shape of types, relative size of types, color of types (?), number relation between types
- compartment level: shape/pos/size/color of compartments / structures
- global level: volumetric dimensions of the data
- todo: what else?

6.5 occlusion

is of course a central factor that could hinder a user's understanding of a transition. can be dealt with at various stages of the pipeline. grouping: choose groups so that they don't interfere with each other (depending on how you plan to transform the data) TODO: come up with examples that describe which kind of grouping would be beneficial for which kind of target state e.g., group the data first into layers parallel to the viewing plane that basically peel the data layouts inner bar: within the bar occlusion does not matter, as long as the bar represents entities of the exact same type todo: other inner layouts & general outer general: the placement of entities/subsets should be chosen so that they don't occlude each other from the viewing angle. maybe we should not distinguish between inner and outer layout since this kind of restricts us to having two layouts instead of n.

then again, we could argue that there can be n-outer layouts, and one inner layout morphing: same as inner layout: within the shape occlusion is not an issue

trajectory: the trajectories offer two of techniques for avoiding occlusion.

edge bundling: setting control points so that paths don't occlude each other todo: anything else?

timing: coordination of timing and speed of elements, e.g., staging

7 CONCLUSION

7.1 future work

implementation of comic strip / animation hybrid

ACKNOWLEDGMENTS

The authors wish to thank A, B, C. This work was supported in part by a grant from XYZ.

REFERENCES

- [1] C. Basch. Animated transitions across multiple dimensions for volumetric data. Master's thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, oct 2011.
- [2] S. Bruckner and M. E. Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, Sept. 2006.
- [3] N. Elmquist, P. Dragicevic, and J. D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, Nov 2008.
- [4] D. Guilmaine, C. Viau, and M. J. McGuffin. Hierarchically animated transitions in visualizations of tree structures. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 514–521, New York, NY, USA, 2012. ACM.
- [5] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, Nov 2007.
- [6] S. Huron, R. Vuillemot, and J.-D. Fekete. Visual Sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, 2013.
- [7] C. Hurter, R. Taylor, S. Carpendale, and A. Telea. Color tunneling: Interactive exploration and selection in volumetric datasets. In *Proceedings of IEEE Pacific Visualization Symposium (PacificVis 2014)*, pages 225–232, March 2014.
- [8] O. Karpenko, W. Li, N. Mitra, and M. Agrawala. Exploded view diagrams of mathematical surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1311–1318, Nov. 2010.
- [9] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. *IEEE Computer*, 46:44–50, 2013.
- [10] R. Kosara, G. N. Sahling, and H. Hauser. Linking scientific and information visualization with interactive 3d scatterplots. In *Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 133–140, 2004.
- [11] W. Li, M. Agrawala, B. Curless, and D. Salesin. Automated generation of interactive 3d exploded view diagrams. *ACM Transaction on Graphics*, 27(3):101:1–101:7, 2008.
- [12] K. L. Ma, I. Liao, J. Frazier, H. Hauser, and H. N. Kostis. Scientific storytelling using visualization. *IEEE Computer Graphics and Applications*, 32(1):12–19, Jan 2012.
- [13] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, Nov 2008.
- [14] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, Nov 2010.
- [15] E. Tufte. *Envisioning Information*. Graphics Press, Cheshire, CT, USA, 1990.
- [16] B. Tversky, J. B. Morrison, and M. Betrancourt. Animation: Can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, Oct. 2002.
- [17] M. Wohlfart and H. Hauser. Story telling for presentation in volume visualization. In *Proceedings of the 9th Joint Eurographics / IEEE VGTC Conference on Visualization*, EUROVIS'07, pages 91–98. Eurographics Association, 2007.