How to ACTUALLY write papers

My practical guide

DK

technologists still believe that the value of their work is independent of whether anyone finds out about it or not. (In any practical discipline, that looks a shaky proposition, even in some mysterious 'moral' sense.) Some of us are still up in Euclidean space, and we need to be talked down.

Like the military, we must learn what soap-manufacturers' sales executives have always known, that the dullest product, produced in quantity, and sold with gusto, will make a much bigger name for its producer than the revolutionary formula which never sees the outside of a test-tube. Technical excellence has to be sold in the same way as soap-powder excellence—not in boxes, but in technical publications—and, yes, these have to be produced in quantity, and sold with gusto. The advertising budgets of the soap conglomerates are enormous, their marketing plans are more elaborate than anything von Schlieffen dreamt of. You must emulate them; allocate half your time to publishing papers, and then read on!

1: Don’t read related work in vain

2: Write systematically

3: Use Overleaf **and** git

4: Macros everywhere

5: Abstract != Introduction, Conclusion != Summary

6: Give your baby a name

7: Use the hell out of Grammarly

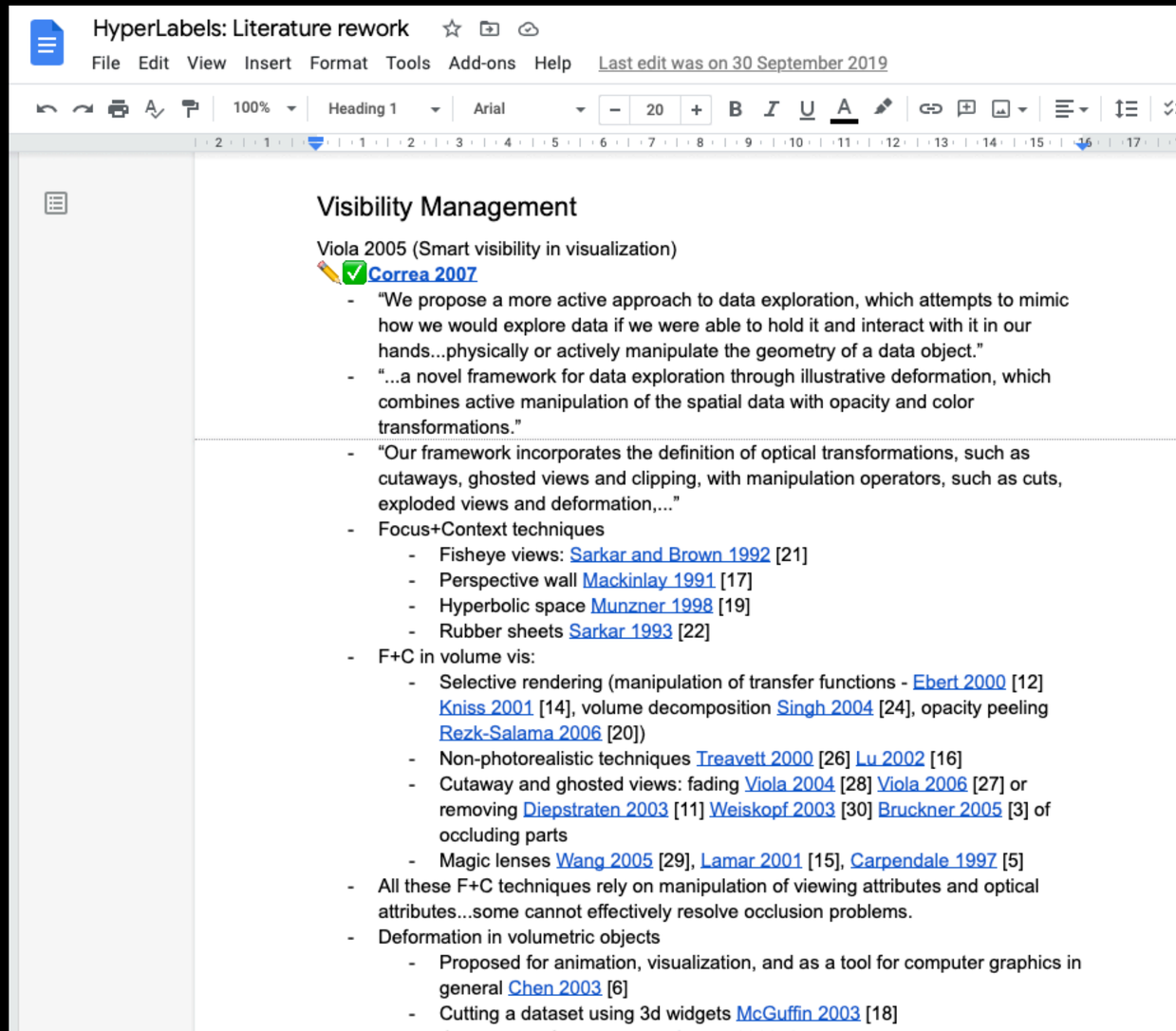
8: Use the hell out of your co-authors

1: Don’t read related work in vain

1: Don’t read related work in vain

- No paper you read should be forgotten
- RW document
  - Taking notes
  - Marking other papers to read
  - Marking finished, reading in progress, etc.
- Remembering papers (Author Year)
- Add all useful papers to the paper (.bib file), group related works

HyperLabelsL Literature rework



2: Write systematically

2: Write systematically

- Don’t be like me, don’t look for the “lowest-hanging fruit”
- Paragraph annotations
- Paragraph states:
  1. First blurt, stream of consciousness
  2. Second pass by me (reduced and readable)
  3. Pass by an external person (making it “official”, in black)

genetic information in the form of a DNA strand, which itself in turn is composed of two chains.

[general hierarchy of abstract structure types - separated from the actual model] This hierarchical organization results in a general hierarchy of structures that abstracts away from any particular dataset. The resulting hierarchy depends on the concrete use case scenario for which it is compiled, and also on the biologist that compiles it. We built our technique on the assumption that this hierarchy can be changed, for several reasons, one of which being that the current knowledge is changing constantly.

[biology can be acquired on different levels] microscopy vs crystalography. ref ML segmentation approach by AICS?

[cell biology vs. molecular biology]

[how we got to this model] truthful biology →concentrations representation (recipe) →structural model

[structure types and structure instances] In order to fit a model like this into a (GPU) memory of a computer, we represent it in a way that considers instances of the structures to be the same. This fits into the concept of instancing, which is used throughout the field of computer graphics. Through instancing, it is possible to not only store, but also render in real-time a models of a size of e-coli bacteria (X millions of atoms).

[structure types hierarchy] Expresses the general organization of abstracted parts (types of structures).

[is it a tree or a graph?]

[instance of higher-level structures] In this paper, we also apply the concept of multiple instances onto higher-level structures. For example, the model of an HIV can be present in the scene several times. In that case for each HIV particle, the type identifier (=HIV) will be the same for each particle, however each will have a unique (relative) instance identifier (0,1,2,...).

[problem: order of encapsulation]

[in practice two types of input] In practice, we work with two inputs. First, the actual **spatial model** which is used for the rendering of the biological structure. Second, the **hierarchical information**, which puts the types of structures into a hierarchical organization, modeling their relationships.

4.1 Structure Type Space

[tree of types]

[interesting here is:] [what children does node have?] [what parent node has?]

4.2 Structure Instance Space

[set of typed instances]

[corresponds to actual rendered objects]

[interesting here is:] [how does the object look?] [where does it fit in the parent] [how many of this object is there (concentration)] [where are the objects placed (distribution)]

5 HyperLabels

[HyperLabels are labels deployed to the scene for easy navigation] HyperLabels are textual labels deployed to the rendered scene in order to facilitate an intuitive navigation in a dense, hierarchical, three-dimensional models.

[they can annotate structures on different levels] First specific on a HyperLabel is that it can annotate objects on a certain level. This way a hierarchy of HyperLabels can be established, with a parent HyperLabel representing a higher-level of annotation of objects labeled by its children.

different behaviour, depending on the semantics. In our case scenario, the different appearance mostly serves to communicate hierarchical relationships between scene objects. Different interactions and behaviours are defined in order to allow intuitive navigation through the hierarchy and 3D model.

[the state depends on what the "current directory" (structure of interest) is] A definition of a "structure of interest" is central to the management of HyperLabels and determining which state a HyperLabel find itself in. A structure of interest is any node in the hierarchy, as described in Section X. This node from the abstract view is tied to the 3D representation by selecting a representative instance for this node. Selection of a structure of interest is done through clicking on a particular HyperLabel.

[what happens when a HyperLabel transitions from one state to another] After this interaction, selected HyperLabel transitions from the "in scene" state to the "context" state and moves to the breadcrumbs panel. This also leads to a change of the set of in scene HyperLabels. Children labels of the picked labels are deployed to the scene while the previously shown labels are removed. This change of state communicates that now we are exploring the structure of interest and are looking at it components.

[labels animations (after click it "flies" to breadcrumbs)]

[they are used to connect two types of information: 3D structural data and abstract hierarchy] Labels in the breadcrumbs panel serve as a contextual information for current 3D view. It communicates where on the spatial hierarchy the user is currently located. This way it connects the 3D spatial data with an abstract meta-data. It enables to examine a detailed view of a certain part of the dataset, while still preserving an idea of where approximately in the data the user is currently located.

[used for navigation by interaction - clicking] Landmarks.

[interaction that we defined only serve as an example] Clicking to switch the current structure of interest is a needed part of the technique in order to use HyperLabels as a way of exploring the dataset. The other interaction presented in this section are only described as an example of features that can be defined in order to fulfill a specific task—in our case the task of navigation, exploration, and most importantly understanding of a hierarchical arrangement of biological 3D structures.

5.1 In Scene State

[hyperlabels in scene serve to connect description with the structure] HyperLabels in the scene state have a primary goal of annotating the rendered 3D structure and through their placement connect the objects with their names. They also serve as entry points for the data, a place where the interaction between the user and the model happens.

5.1.1 Appearance

- additional widgets around the label?
- to indicate which are important (HIV is more interesting to click than Albumin)
- to give a scale
- how to indicate "clickability"?

[2D labels overlaid over the final render] We render the HyperLabels in the scene state and 2D billboards overlaid over the final rendered frame.

for a HyperLabel hovered over, to indicate, together with the changed cursor, that such text is clickable.

cognitive load, as the user does not have to switch contexts and shift focus between multiple views to get the presently relevant information.

[linking of interactions in hierarchy view and 3D view] Interactions can happen both in the spatial view, as well as in the integrated abstract view. The definition of the current structure of interest links both views. No matter through which view the current structure in focus is selected, both views need to be altered to reflect this change. This linkage is essential to the core idea of our method, where the spatial view presents possibilities for traversing further in the object hierarchy while the abstract view holds the path to the current structure in focus and enables traversing the hierarchy in the opposite direction.

4.2 HyperLabels

[combine annotation with navigation] HyperLabels placed in the 3D spatial view still fulfill the role of traditional textual labels—they connect the rendered 3D structures with their annotations through their placement. On top of this, they are an entry point for the data, a place where the interaction between the user and the model can happen. Without labels, selecting a structure of interest would be ambiguous because of its multi-scale character: a click anywhere in the scene can mean selection of any level of the shown structure. Therefore, labels are a great means for defining a specific small screen region where an interaction is conducted and the selection task is executed.

[our case: also support task of navigation through this concept] In our proposed method, HyperLabels lead also to a solution of the navigation task. This is executed in conjunction with the Structure of Interest Opening, where after a structure of interest is selected by clicking on an associated HyperLabel, the structure's presentation is adjusted as well as the camera parameters.

[clicking on labels is better than zooming] This new way of navigating 3D environments is needed because of the shortcomings of traditional navigation metaphors in biological scenes, which feature structures spanning several magnitudes of scale, are incredibly dense, and truly fill 3D space. Navigational methods for the macroscale usually utilize three interactions: orbiting around the model, panning, and zooming (flying in the camera direction). Since we aim at visualizing vast multi-scale scenes, moving between structures of significantly different sizes would lead to a frequent use of the zooming functionality. It is difficult to set the appropriate zooming speed as it has to be adjusted according to the scale level. For scales representing a whole virus, the desirable speed is higher than the one for precisely would provide a significant benefit, as in that case the process of transitioning between two salient objects would be automated and would require no further user interaction. HyperLabels are an implementation of such a concept. They also provide a certain degree of guidance—thanks to repurposing of textual labels—and, at the same time, present different options for landmarks worth exploring and flying towards.

[HyperLabels serve to establish structure of interest] The primary purpose of interaction on HyperLabels is to establish the current structure in focus. A click on a HyperLabel selects the corresponding object as the current structure of interest and the visualization is altered to display this level of the model. We however see the potential of HyperLabels to serve as a base framework for future extensions. Additional interactions can be defined and implemented, to carry specific functionality and further support users in exploring and understanding a hierarchical 3D model (see subsection 7.2).

[why the name] We see several parallels of HyperLabels with the concept of hyperlinks from the Web. When a hyperlink is clicked, it transports the user to a new location (a web page), and the content of this new location is shown in the form of text, images, and other multimedia. HyperLabels work in a similar fashion: different locations (landmarks) are used as the destinations to which users are transported, and the content in our case is the inner structure, i.e., subparts of landmarks. We have adopted the name, as well as some design concepts, in the proposed active and navigational labels which we term *HyperLabels*.

[designed for mouse but can be also touch] We designed, implemented, and tested our method with a classic mouse interface, where we used the left button for clicking. We use the terminology of this environment in our description, but from the conceptual standpoint, adapting the technique for use on touch interface or other similar setups should be straightforward.

We render HyperLabels as 2D billboards superimposed over the final rendered frame.

[hover over] Since labels in 3D visualization are traditionally used in a passive way, viewers might not even realize that such action as clicking on the label is possible. There are several options for indicating that a HyperLabel can be clicked. We decided to follow the convention established in web environments: we change the visual appearance upon hover—by modifying the color of the text and rendering a background box for the hovered HyperLabel—and we also change the cursor shape when it is positioned over a HyperLabel.

4.3 Breadcrumbs Panel

HyperLabels in the breadcrumbs panel provide context for current location and enable traversal of the hierarchy in

Don’t be affraid to REARRANGE

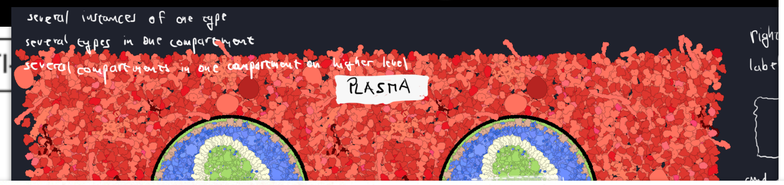
What you wrote so far is not lost if you maintain paragraphs as the atomic elements

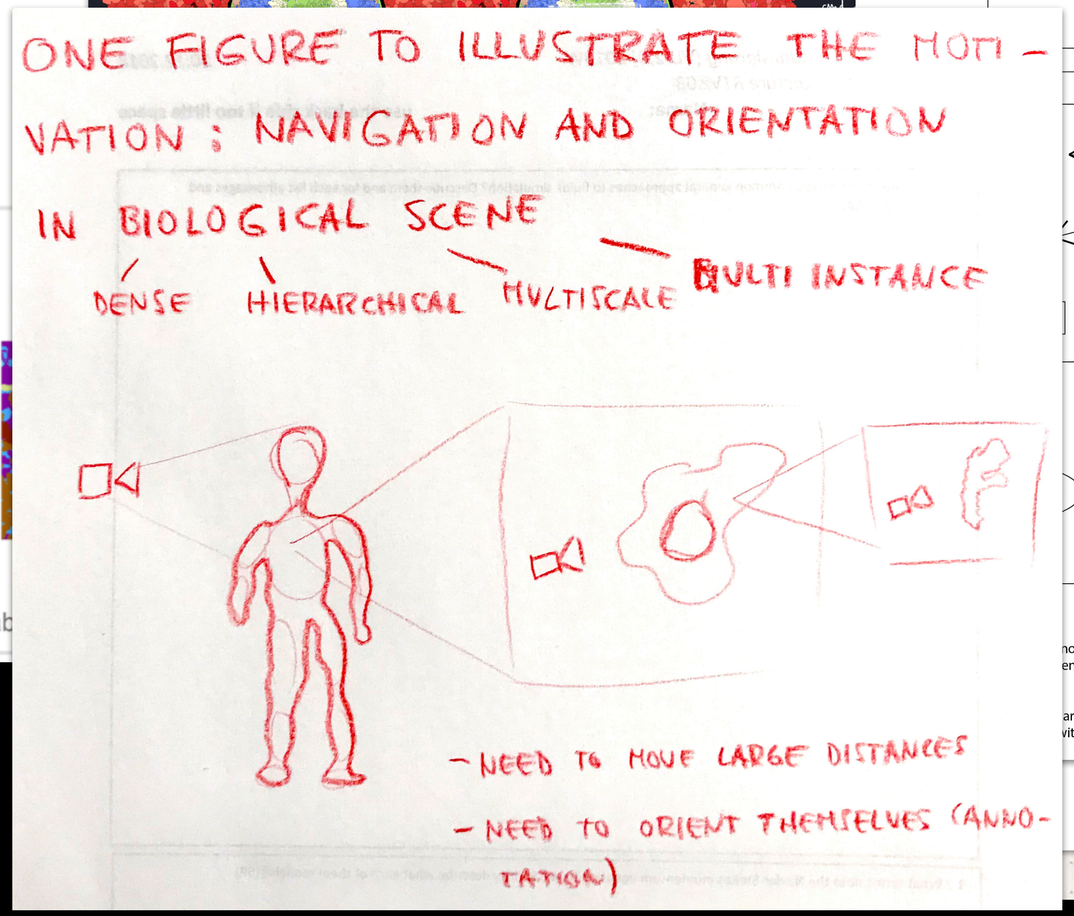
CONNECT your paragraphs

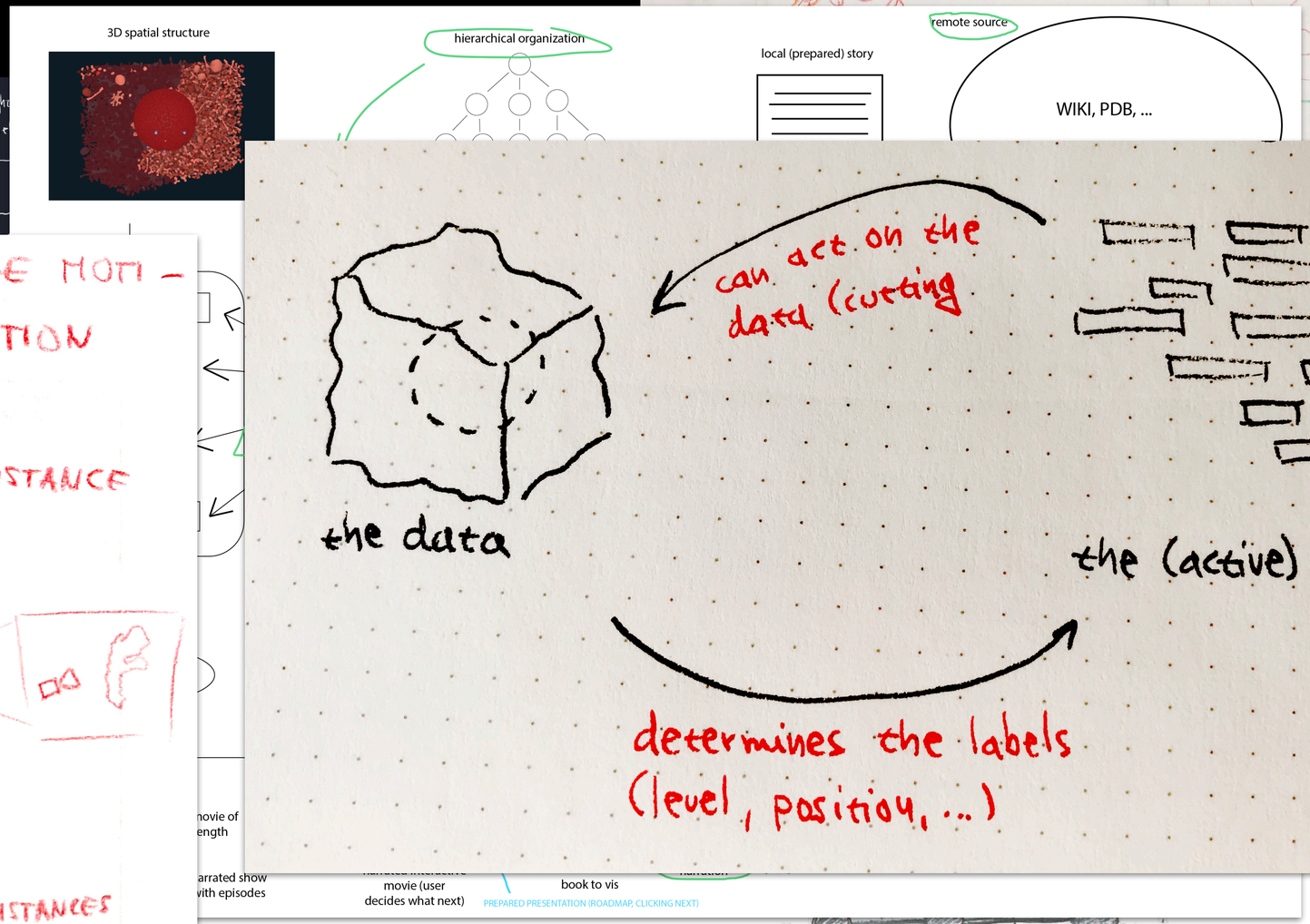
Don’t just state things, make links between them

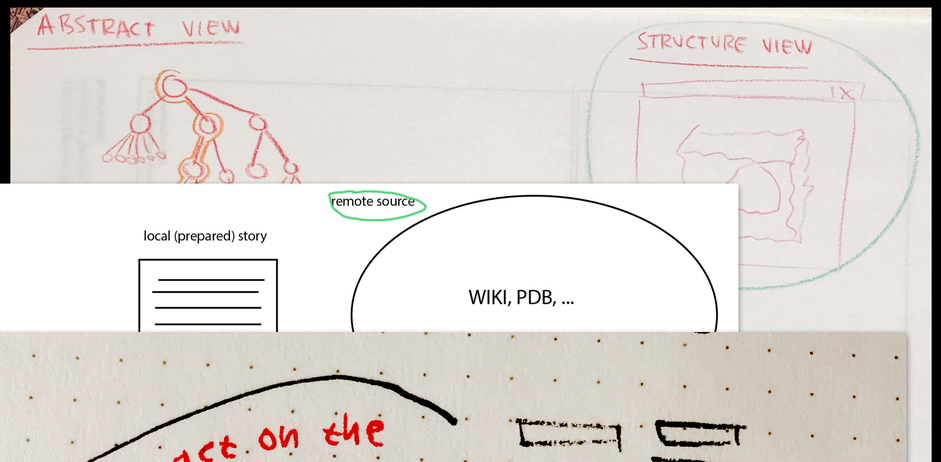
2: Write systematically

Overview Figures









using HyperLabels: Grey box indicates the traditional annotation pipeline, in which textual labels play only a passive role.
resented in the paper—sparsification, anchoring, and re-annotation—turn the labels into active elements (HyperLabels)
through which an intuitive exploration of all levels of hierarchy of a model is possible.

3: Use Overleaf and git

3: Use Overleaf and git

- Helps eliminate errors (especially in later parts of writing)
- Helps you write systematically
- Can use your favorite editor (for me: Emacs)
- Local TeX setup: check for compile errors before pushing
- Setup:
  - Create project on Overleaf first
  - Separate .tex files into sections
  - On opened Overleaf project: Menu -> Sync: git -> clone
  - .gitignore from: github.com/github/gitignore/blob/main/TeX.gitignore

4: Macros everywhere

4: Macros everywhere

```
\newcommand{\eg}{e.\,g.}
\newcommand{\ie}{i.\,e.}

\newcommand{\dknote}[1]{\marginpar{\footnotesize\textcolor{BlueViolet}{#1}}}
\newcommand\delcand[1]{\textcolor{Gray}{#1}}


\newcommand\dk[1]{\textcolor{BlueViolet}{#1}}
\newcommand\os[1]{\textcolor{RedOrange}{#1}}
\newcommand\iv[1]{\textcolor{NavyBlue}{#1}}
\newcommand\ti[1]{\textcolor{BrickRed}{#1}}


\newcommand{\annot}[1]{
\iftrue % uncomment to SHOW paragraph annotations
%\iffalse % uncomment to HIDE paragraph annotations
\textbf{[#1]}
\fi
}
```

5: Abstract != Introduction, Conclusion != Summary

5: Abstract != Introduction, Conclusion != Summary

- Introduction must tell a story

> *“Currently, the state is this. If only we could solve X, then Y, Z, W would be possible. But X hasn’t been solved before. In this paper, we propose a solution to X. We contribute: …”*
>
> OR
>
> *“What is…And yet…But what could be…Here’s how we do this.”*

- Keep it “short and sweet” (not longer than page 1, if possible)
- Abstract = most concise summary for expert readers
- Conclusion: come back to introduction → are the mentioned problems solved? What are the implications?

5: Abstract != Introduction, Conclusion != Summary

Abstract Formula

- WHAT: precisely state what you are presenting (No “We work with xyz data”)
- WHY: give motivation
- HOW: how did you accomplish to solve this problem
- RESULTS: specific results achieved using your solution

Before

HyperLabels: A Framework for Browsing Dense Hierarchical Multi-Instance and Multi-Scale 3D Biological Models

David Kouřil & friends

Abstract— Interactivity is an important aspect in visualization, as it enables to influence the way data are visually presented with an immediate feedback to the user. Navigation in 3D space is a prime example of interaction in scientific visualization. More and more phenomena with inherently multi-scale character—like astronomical, geographical, or biological data—need to be visualized and navigated through. Models coming from structural biology bring additional challenge—an incredible crowdedness, to a degree where only the outer layer of the model is visible unless a cut away approach is used. Often these models are hierarchically organized and in order to truly appreciate and understand the particular phenomena, this hierarchy needs to be communicated, ideally still with reference to the actual spatial model. Finally, in multi-scale scenes representing structures on various scales, the problem of examining such scenes from the atomic level to a whole virus, or possibly even a whole cell, arises. In this paper we present the HyperLabels framework which addresses these challenges based on three key insights. First, we define a new kind of textual label—HyperLabel—on which a specific interaction can be defined. Through this interaction a current Structure of Interest (SoI, i.e., a selected object in the scene) is indicated. Second, we take the hierarchical information about the model into account when resolving cut away settings for a current SoI. Finally, by keeping a history of HyperLabel interactions, we use this as a contextual information to help users build up a mental image (understanding) of the hierarchical structure of the model. This is achieved by incorporating a breadcrumbs panel—a concept well-known in web design. We present the framework around HyperLabels by introducing its three core parts: Visibility management, HyperLabels management, and Camera management. We show how HyperLabels can be used to browse a

After

HyperLabels: Browsing Dense and Hierarchical Molecular 3D Models

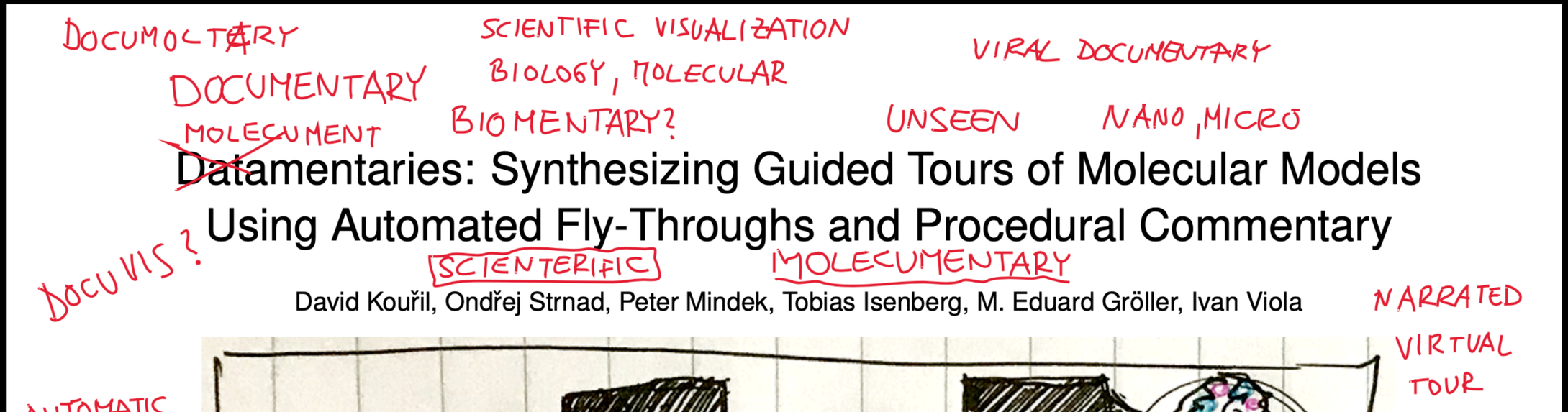
David Kouřil & friends

Abstract—We present HyperLabels for exploration of hierarchical 3D datasets. A HyperLabel is a new type of textual label, which, as an active element, supports both navigation as well as annotation. The need for this novel concept stems from visualization of multi-scale molecular data. Unlike other complex multi-scale phenomena, e. g., from astronomy or geography, models from structural biology in addition are densely crowded. Cut-aways are needed to reveal the inner organization; otherwise, only the outer layer of the model is visible. The typically hierarchical organization of these models needs to be communicated to fully convey and understand the particular structure, ideally with reference to the actual spatial model. HyperLabels are interactive and intuitively facilitate the navigation of multi-scale scenes. We accomplish this in three steps: (1) through interaction with HyperLabels the current structure of interest is selected; (2) we employ the hierarchical information in the model to determine cut-away settings for the current focus target; (3) we visualize the current location in the hierarchy in a breadcrumbs panel, a concept well-known in web design. The viewers are able to explore the multi-scale model from the highest level (e. g., a virus) to the lowest one (e. g., atoms). In this way, we simultaneously communicate both the abstract information (hierarchy) and the spatial information (3D structure). We demonstrate how HyperLabels are used to browse two exemplary models from meso-scale biology where both the spatial and abstract components are explored.

6: Give your baby a name

6: Give your baby a name

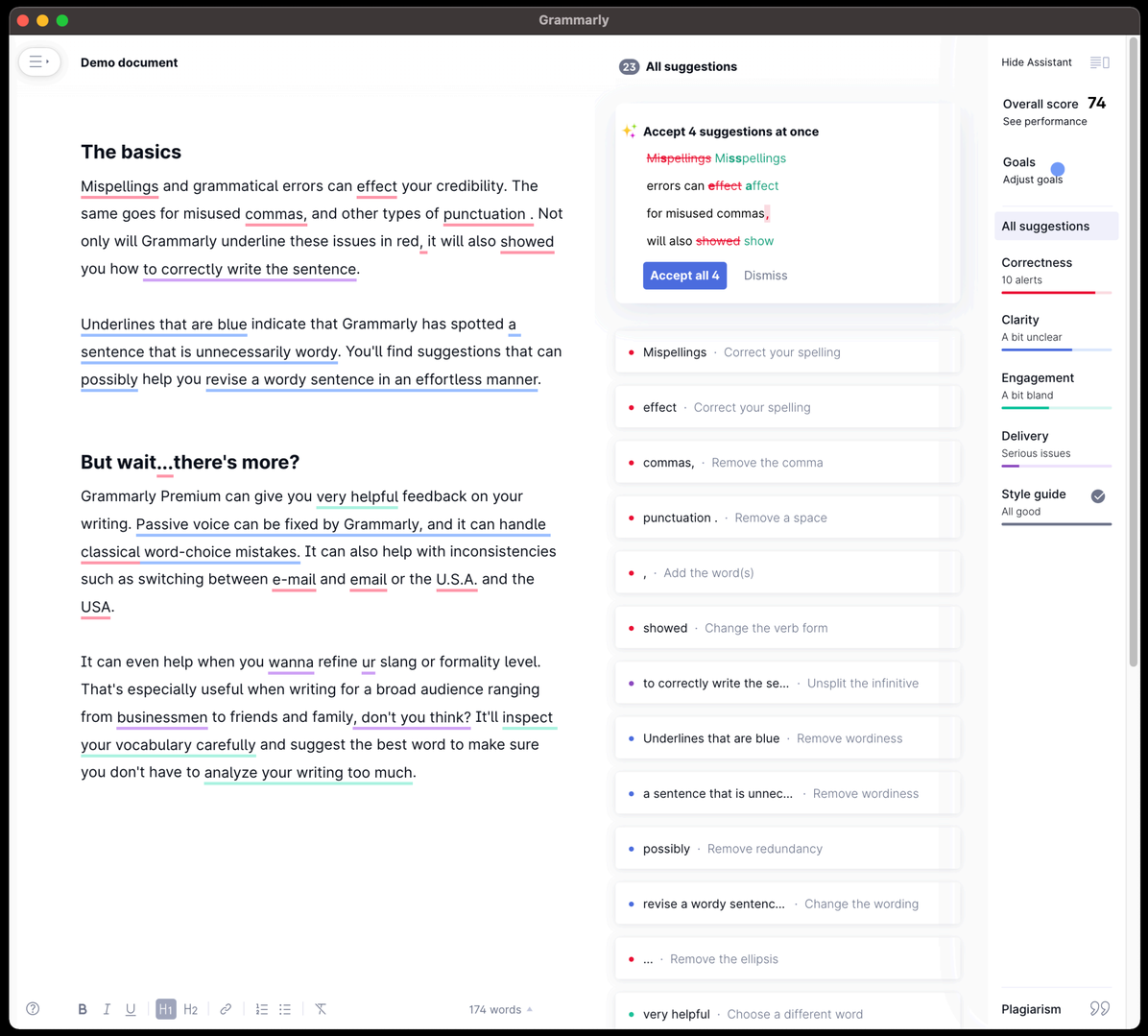
- Paper title
- Parts of your technique
- Conceptual (e.g., Anders Ynnerman's *Exploranation*)
- 70% of paper writing (for me) = sketches + naming



7: Use the hell out of Grammarly

7: User the hell out of Grammarly

- Non-native speakers = disadvantage
- Grammar checking but also sentence structure, word combinations, …
- Also other tools:
  - capitalizemytitle.com
  - thesaurus.com (synonyms)



8: Use the hell out of your co-authors

TI Tobias, 15:30

Pass on 1 done, releasing lock. Generally nice: just make sure that you ALWAYS use active voice, no excuses allowed.

Meister has some good comments, but you do not need to incorporate all of them. In particular make sure that you do not generate new orphans.

General note: try to work on your use of passive voice. Essentially try to avoid it wherever possible Always write "we do"

0:44

oh wowo tobias thank you so much for going through the whole paper

TI Tobias, 0:44

Well, I am a co-author, ain't I?

0:44

you for sure are

TI Tobias, 0:45

And that means for me that I read, edit, and comment on a paper before it goes in.

You have to make the connection for virtually anything you mention, otherwise why should you mention it?

I.e.: what aspects do you use? what aspects are limited for your application case? where do you borrow ideas or where are solutions similar but you use a different technique, etc.

8: Use the Hell Out of Your Co-authors

- Different people will give you different things (design stage, implementation, direction, writing, proof-reading, …even career advice)
- Plan so that you use the right people at the right time
- You are the almighty co-ordinator → you make the decision

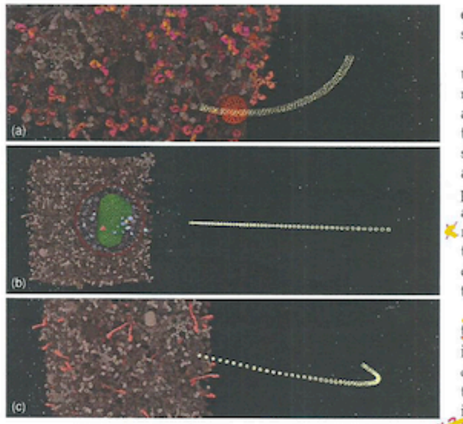


Fig. 7. Illustration of the three camera animation types used in a molecumentary synthesis: anchored orbiting (a), direct flying (b), and curved transition (c).

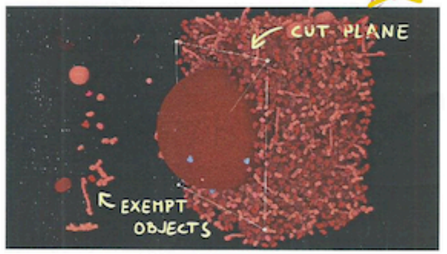


Fig. 8. Traveling cutting plane: We remove all objects—except a selected subset—that lie between the cutting plane and the camera position to reveal inside components of the model.

For a continuous narrative, however, we also need to transition between two focus instances, for which we use *direct flying*. We animate the camera along a straight line, with its orientation fixed. This movement type is suitable for cases where the two instances (initial and target) are visible from the initial camera viewpoint. If the target position is outside of the viewing frustum, direct flying can be suboptimal in communicating the spatial relation between the two objects.

Therefore, we introduce the third movement type *curved path animation*. In this animation type we zoom the camera slightly out of the initial focus position, providing context of its surroundings, and then travel toward the target focus position on a curved path. We use a quadratic Bézier curve, but any curve type can be used.

We apply easing functions to the camera transitions for a smoother impression and visually more pleasing movement.

5.1.2 Occlusion Management

Biological models are densely packed with molecules, which results in occlusion of most of the interesting structures, e. g., inside of a virus. Occlusion management is required to showcase all relevant parts of the model properly.

We employ a *traveling cutting plane* approach (see Figure 8). We define a cutting plane in the scene and do not render any object that lies between the cutting plane and the camera. We exclude, however, certain instances (or types) from being cut away. This allows us to highlight the selected objects as well as convey the impression of the absolute number of these objects in the model. The cutting plane *travels*, i. e., we animate it and the set of objects we always show throughout the molecumentary, to successively reveal objects that are being verbally described. We perform these animated transitions in the *transition scenes*. We then determine the objects exempt from removal based on the type of the scene that follows the transition.

For a *focus scene*, we shift attention to one (sub-)structure type. To emphasize this focus type, we exempt all its instances from being cut for the duration of the scene to communicate their number in the model. We then re-position the cutting plane to the center of a selected representative instance. We select the instance closest to the camera as the representative and orient the cutting plane to be parallel to the viewing plane at the moment the object comes into focus, i. e., we orient it according to the camera's initial back vector.

An *overview scene* communicates inner composition of a structure. The *transition scene* leading up to an overview scene features an animation that opens up the structure of interest and reveals its inside. We do so by fetching the structural components (child nodes) of the focus structure and, for each of these child nodes, pick a representative instance and exempt it from the cutting. We then place the cutting plane at the position of the representative that is furthest away from the camera such that none of the representatives is occluded by instances kept in the scene.

We purposefully used the traveling cutting plane as a world-space technique that culls instances, rather than image blending effects. The fading in and out of alpha blending resembles a "cut" in movie making, which could make it less apparent that our scene changes communicate an opening of the model, as opposed to a change of the scene altogether. The added complexity of managing several cutting planes to ensure that an object selected in the future will not be cut away, however, in our opinion outweighs the potential benefits. We also only use a single cutting plane in our design to avoid the complexity of managing multiple planes of even a plane hierarchy: It would be difficult to ensure that an object, selected later in the molecumentary, is not cut away.

5.1.3 Verbal Commentary

We realize the verbal commentary using text-to-speech synthesis. We assemble three types of commentary—structural, descriptional, and navigational—in textual form first and then turn them into speech using an artificial voice.

We use *structural commentary* in overview scenes to describe the structural composition of certain composite objects. An example of structural commentary is *"Blood plasma consists of Hemoglobin and Heparin and others."* We construct the commentary procedurally based on the hierarchical object composition using sentence templates. We define basic sentence templates in an external file which can be further extended. Our basic templates communicating hierarchical

Almost The End.

Other Sources

- Design: The Key to Writing (and Advising) a One-Draft Thesis (John Carlis)
- Common mistakes in technical writing (Wojciech Jarosz)

The End.

1: Don’t read related work in vain

2: Write systematically

3: Use Overleaf **and** git

4: Macros everywhere

5: Abstract != Introduction, Conclusion != Summary

6: Give your baby a name

7: Use the hell out of Grammarly

8: Use the hell out of your co-authors