Process and Pitfalls in Writing Information Visualization Research Papers

Tamara Munzner

University of British Columbia
tmm@cs.ubc.ca, http://www.cs.ubc.ca/∼tmm

Abstract. The goal of this paper is to help authors recognize and avoid a set of pitfalls that recur in many rejected information visualization papers, using a chronological model of the research process. Selecting a target paper type in the initial stage can avert an inappropriate choice of validation methods. Pitfalls involving the design of a visual encoding may occur during the middle stages of a project. In a later stage when the bulk of the research is finished and the paper writeup begins, the possible pitfalls are strategic choices for the content and structure of the paper as a whole, tactical problems localized to specific sections, and unconvincing ways to present the results. Final-stage pitfalls of writing style can be checked after a full paper draft exists, and the last set of problems pertain to submission.

1 Introduction

Many rejected information visualization research papers have similar flaws. In this paper, I categorize these common pitfalls in the context of stages of the research process. My main goal is to help authors escape these pitfalls, especially graduate students or those new to the field of information visualization. Reviewers might also find these pitfalls an interesting point of departure when considering the merits of a paper.

This paper is structured around a chronological model of the information visualization research process. I argue that a project should begin with a careful consideration of the type of paper that is the desired outcome, in order to avoid the pitfalls of unconvincing validation approaches. Research projects that involve the design of a new visual encoding would benefit from checking for several middle-stage pitfalls in unjustified or inappropropriate encoding choices. Another critical checkpoint is the late stage of the project, after the bulk of the work is done, but before diving in to writing up results. At this point, you should consider both strategic pitfalls about the high-level structure of the entire paper, tactical pitfalls that affect one or a few sections, and possible pitfalls in the specifics of your approach to the results section. At a final stage, when there is a complete paper draft, you can check for lower-level pitfalls of writing style, and avoid submission-time pitfalls.

I have chosen a breezy style, following in the footsteps of Levin and Redell [22] and Shewchuk [34]. My intent is serious, but I have tried to invent catchy – sometimes even snide – titles in hopes of making these pitfalls more memorable. Guides to writing research papers have been written in several subfields of computer science, including systems [22], software engineering [33], programming languages [19], networking [28], and graphics [20]. Many of the pitfalls in the middle and later project stages apply to research writing in general, not just information visualization, and have been mentioned in one or many of these previous papers.

My first pass at providing advice for authors and reviewers in the field of information visualization, abbreviated as **infovis**, was the creation of the author guide for the annual conference. When I was Posters Chair of InfoVis 2002, the IEEE Symposium on Information Visualization, I read the roughly 300 reviews of the 78 rejected papers in order to decide which to invite as poster submissions. The experience convinced me that future paper authors would benefit from more specific guidance. When I became Papers Chair in 2003, with co-chair Stephen North, we completely rewrote the Call for Papers. We introduced five categories of papers, with an explicit discussion of the expectations for each, in a guide for authors that has been kept unchanged through the 2007 conference.

This second pass is motivated by the patterns of mistakes I saw in my two-year term as InfoVis Papers Co-Chair where I read the over 700 reviews for all 189 submitted papers, and in personally writing nearly 100 reviews in the subsequent three years. My discussion of paper types below expands considerably on the previous author guide, and I provide concrete examples of strong papers for each type. The advice I offer is neither complete nor objective; although I draw on my experience as a papers chair, my conclusions may be idiosyncratic and reflect my personal biases. I do not perform any quantitative analysis. Doing so in the domain of infovis would, no doubt, be fruitful future work, given the interesting results from software engineering [33] and human-computer interaction [17].

None of these pitfalls are aimed at any particular individual: I have seen multiple instances of each one. Often a single major pitfall was enough to doom a paper to rejection, although in some cases I have seen other strengths outweigh a particular weakness. I hasten to point out that I, myself, have committed some of the errors listed below, and despite my best efforts I may well fall prey to them in the future.

2 Initial Stage: Paper Types

A good way to begin a research project is to consider where you want it to end. That is, what kind of a paper do you intend to produce? That choice should guide many of your downstream decisions, including the critical issue of how to validate any claims you might make about your research contribution.

2.1 Validation Approaches

Many possible ways exist to validate the claim that an infovis paper has made a contribution, including:

– algorithm complexity analysis
– implementation performance (speed, memory)
– quantitative metrics
– qualitative discussion of result pictures
– user anecdotes (insights found)
– user community size (adoption)
– informal usability study
– laboratory user study
– field study with target user population
– design justification from task analysis
– visual encoding justification from theoretical principles

In any particular paper, the constraints of researcher time and page limits force authors to select a subset of these approaches to validation. The taxonomy of paper types below can provide you with considerable guidance in choosing appropriate validation approaches, leading to a paper structure where your results back up your claims. The five paper types guide the presentation of your research by distinguishing between the following possibilities for your primary contribution: an algorithm, a design, a system, a user study, or a model.

2.2 Technique

Technique papers focus on novel algorithms and an implementation is expected. The most straightforward case is where the research contribution is a new algorithm that refines or improves a technique proposed in previous work. A typical claim is that the new algorithm is faster, more scalable, or provides better visual quality than the previously proposed one. The MillionVis system [5], hierarchical parallel coordinates [6], and hierarchical edge bundling [15] are good exemplars for this category.

Typical results to back up such a claim would be algorithm complexity analysis, quantitative timing measurements of the implementation, and a qualitative discussion of images created by the new algorithm. Quantitative metrics of image quality, for example edge crossings in graph layout, are also appropriate. You need to compare these results side by side against those from competing algorithms. You might collect this information through some combination of using results from previous publications, running publicly available code, or implementing them yourself. In this case, there is very little or no design justification for whether the technique is actually suitable for the proposed problem domain in the paper itself: there is an implicit assumption that the previous cited work makes such arguments.

In retrospect, a better name for this category might be **Algorithms**. Many authors who design new visual representations might think that a paper documenting a new technique belongs in the Technique category. However, the question to ask is whether your primary contribution is the algorithm itself, or the design. If your algorithm is sophisticated enough that it requires several pages of description for replicability, then you probably have a primary algorithmic contribution. If the algorithm itself is straightforward enough that only a brief description is required, or if all of the techniques that you use have been adequately described in previous work, then you would be better served by explicitly writing a Design Study paper.

2.3 Design Study

Design Study papers make a case that a new visual representation is a suitable solution for a particular domain problem. First, you should explain the target problem. You must provide enough background that the reader can pass judgement about whether your solution is good, but not so much detail that the focus of the paper is on domain problems rather than infovis issues. Finding the right balance is a difficult but crucial judgement call. Second, you should crisply state the design requirements that you have determined through your task analysis. Third, you should present your visual encoding and interaction mechanisms and justify these design choices in terms of how well they fulfill the requirements. Typical arguments would refer to perceptual principles and infovis theory. For example, using spatial position to encode the most important variables and using greyscale value rather than hue to encode an ordered variable are both very defensible choices [24]. The best justifications explicitly discuss particular choices in the context of several possible alternatives.

Fourth, you should present results that back up the claim that your approach is better than others. Typical results include case studies or scenarios of use. Design studies often document iterative design and the use of formative evaluation for refinement. The research contribution of a design study is not typically a new algorithm or technique, but rather a well-reasoned justification of how existing techniques can be usefully combined. For most design studies, adoption by the target users is valuable evidence that the system has met its goals, as are anecdotes of insights found with the new system that would be difficult to obtain using previous methods.

I think this category name is still a good choice, despite the fact that great design studies are all too rare. I argue that the field would be well served if more authors explicitly cast their work in this category. Interesting examples that approach the design study from several angles are the cluster-calendar system [40], ThemeRiver [11], Vizster [14], VistaChrom [21], and a hotel visitation pattern analysis system [42].

2.4 Systems

Systems papers focus on the architectural choices made in the design of an infrastructure, framework, or toolkit. A systems paper typically does not introduce new techniques or algorithms. A systems paper also does not introduce a new design for an application that solves a specific problem; that would be a design study. The research contribution of a systems paper is the discussion of architectural design choices and abstractions in a framework or library, not just a single application. A good example is the prefuse systems paper [13], which has a discussion of the performance, flexibility, and extensibility implications of the ItemRegistry, Action, and ActionList data structure choices. Another good example is the systems paper on design choices made in Rivet [37] and other systems with similar goals, such as the tradeoffs of data granularity for transformations.

A systems paper can be considered as a specialized kind of design study: one about the choices made when building a library as opposed to the choices made when solving a visual encoding problem. Like the design study category, key aspects of a systems paper are the lessons learned from building the system, and observing its use. I urge authors and reviewers of systems papers to peruse Levin and Redell's classic on "How (and How Not) to Write a Good Systems Paper" [22].

The category name might be a cause for confusion because the the term system is often used interchangeably with application or implementation. The original intent was to follow the distributed systems usage where there is a very strong distinction between system-level and application-level work. Although a name like **Toolkit** might avert that confusion, the term 'systems paper' is such a strong convention in computer science that I am reluctant to advocate this change.

2.5 Evaluation

Evaluation papers focus on assessing how an infovis system or technique is used by some target population. Evaluation papers typically do not introduce new techniques or algorithms, and often use implementations described in previous work. The most common approach in infovis thus far has been formal user studies conducted in laboratory setting, using carefully abstracted tasks that can be quantitatively measured in terms of time and accuracy, and analyzed with statistical methods. A typical claim would be that the tested tasks are ecologically valid; that is, they correspond to those actually undertaken by target users in a target domain. A typical result would be a statistically significant main effect of an experimental factor, or interaction effect between factors. The work of Yost and North on perceptual scalability is a good example of this subtype [44]. A different approach to studying user behavior is field studies, where a system is deployed in a real-world setting with its target users. In these studies, the number of participants is usually smaller, with no attempt to achieve statistical significance, and the time span is usually weeks or months rather than hours. However, the study design axes of field versus laboratory, short-term versus long-term, and size are all orthogonal. Both quantitative and qualitative measurements may be collected. For example, usage patterns may be studied through quantitative logging of mouse actions or eyegaze. The work of Hornbæk and Hertzum on untangling fisheye menus is a good example of this subtype [16]. Usage patterns can also be studied through qualitative observations during the test itself or later via coding of videotaped sessions. Trafton *et al.*'s field study of how meteorologists use visual representations is an excellent example of the power of video coding [39].

Plaisant's discussion of the difficulties of evaluating infovis is thoughtful and germane [29]. In retrospect, a better, albeit wordy, name for this category might be **Summative User Studies**, since the goal is to examine the strengths of a system or technique. Evaluation is far too broad a term – because all papers should contain some sort of validation. Even User Studies would not be the best choice, because formative studies are probably a better fit for the Design Study category, where ethnographic methods are often appropriate in the task analysis to determine design requirements or iteratively refine a design. However, these lines are not necessarily crisp. For instance, the MILC approach advocated by Shneiderman and Plaisant [35] could fit into either a Design Study framework, if the emphasis is on the formative ethnographic analysis and iterative design, or a Summative Evaluation framework, if the emphasis is on the longitudinal field study.

2.6 Model

Model papers present formalisms and abstractions as opposed to the design or evaluation of any particular technique or system. This category is for meta-research papers, where the broad purpose is to help other researchers think about their own work.

The most common subcategory is **Taxonomy**, where the goal is to propose categories that help researchers better understand the structure of the space of possibilities for some topic. Some boundaries will inevitably be fuzzy, but the goal is to be as comprehensive and complete as possible. As opposed to a survey paper, where the goal is simply to summarize the previous work, a taxonomy paper proposes some new categorization or expands upon a previous one and may presume the reader's familiarity with the previous work. Good examples are Card and Mackinlay's taxonomy of visual encodings [3] and Amar *et al.*'s task taxonomy [1].

A second subcategory is **Formalism**, for papers that present new models, definitions, or terminology to describe techniques or phenomena. A key attribute of these kinds of papers is reflective observation. The authors look at what is going on in a field and provide a new way of thinking about it that is clear, insightful, and summative. An influential example is the space-scale diagram work of Furnas and Bederson [7], and an interesting recent example is the casual infovis definition from Pousman *et al.* [31].

A third subcategory is **Commentary**, where the authors advocate a position and argue to support it. Typical arguments would be "the field needs to do more X", "we should be pushing for more Y", or "avoid doing Z because of these drawbacks". A good example is the fisheye followup from Furnas [8]. These kinds of papers often cite many examples and may also introduce new terminology.

Model papers can provide both a valuable summary of a topic and a vocabulary to more concisely discuss concepts in the area. They can be valuable for both established researchers and newcomers to a field, and are often used as assigned readings in courses. I think this category name is appropriate and do not suggest changing it.

2.7 Combinations

These categories are not hard and fast: some papers are a mixture. For example, a design study where the primary contribution is the design might include a secondary contribution of summative evaluation in the form of a lab or field study. Similarly, a design study may have a secondary contribution in the form of a novel algorithm. Conversely, a technique paper where the primary contribution is a novel algorithm may also include a secondary design contribution in the form of a task analysis or design requirements. However, beware the *Neither Fish Nor Fowl* pitfall I discuss below.

2.8 Type Pitfalls

Carefully consider the primary contribution of your work to avoid the pitfalls that arise from a mismatch between the strengths of your project and the paper type you choose.

Design in Technique's Clothing: Don't validate a new design by providing only performance measurements. Many rejected infovis papers are bad design studies, where a new design is proposed but the design requirements are not crisply stated and justification for design choices is not presented. Many of these authors would be surprised to hear that they have written a design study, because they simply assume that the form of a technique paper is always the correct choice. Technique papers are typically appropriate if you have a novel algorithmic contribution, or you are the very first to propose a technique. If you are combining techniques that have been previously proposed, then the design study form is probably more appropriate.

Application Bingo versus Design Study: Don't apply some random technique to a new problem without thoroughly thinking about what the problem is, whether the technique is suitable, and to what extent it solves the problem. I define 'application bingo' as the game where you pick a narrowly defined problem domain, a random technique, and then write an application paper with the claim of novelty for this particular domain-technique combination. Application bingo is a bad game to play because an overwhelming number of the many combinatorial possibilities lead to a bad design.

Although application bingo is admittedly a caricature, the important question is how we can distinguish those who inadvertently play it from those who genuinely solve a domain problem with an effective visualization. Some visualization venues distinguish between research papers and what are called applications or case studies. This paper category is often implicitly or explicitly considered to be a way to gather data from a community outside of visualization itself. Although that goal is laudable, the mechanism has dangers. A very common pitfall is that application paper submissions simply describe an instantiation of a previous technique in great detail. Many do not have an adequate description of the domain problem. Most do not have an adequate justification of why the technique is suitable for the problem. Most do not close the loop with a validation that the proposed solution is effective for the target users.

In contrast, a strong design study would be rather difficult for an outsider unfamiliar with the infovis literature to write. Two critical aspects require a thorough understanding of the strengths and weaknesses of many visualization techniques. First, although a guideline like "clearly state the problem" might seem straightforward at first glance, the job of abstracting from a problem in some target domain to design requirements *that can be addressed through visualization techniques* requires knowing those techniques. Second, justifying why the chosen techniques are more appropriate than other techniques again requires knowledge of the array of possible techniques.

The flip side of this situation is that design studies where visualization researchers do not have close contact with the target users are usually also weak. A good methodology is collaboration between visualization researchers and target users with driving problems [18](Chapter 3.4).

All That Coding Means I Deserve A Systems Paper: Many significant coding efforts do not lead to a systems paper. Consider whether or not you have specific architectural lessons to offer to the research community that you learned as a result of building your library or toolkit.

Neither Fish Nor Fowl: Papers that try to straddle multiple categories often fail to succeed in any of them. Be ruthlessly clear about identifying your most important contribution as primary, and explicitly categorize any other contributions as secondary. Then make structural and validation choices based on the category of the single primary contribution.

3 Middle Pitfalls: Visual Encoding

If you have chosen the design route, then a major concern in the middle stages of a project should be whether your visual encoding choices are appropriate and justifiable.

Unjustified Visual Encoding: An infovis design study paper must carefully justify why the visual encoding chosen is appropriate for the problem at hand. In the case of technique papers, where the focus is on accelerating or improving a previously proposed technique, the argument can be extremely terse and use a citation to a previous paper. But in the case of a design study, or a paper proposing a completely new technique, your justification needs to be explicit and convincing. One of the most central challenges in information visualization is designing the visual encoding and interaction mechanisms to show and manipulate a dataset.

A straightforward visual encoding of the exact input data is often not sufficient. In many successful infovis approaches, the input data undergoes significant transformations into some derived model that is ultimately shown. Many weak papers completely skip the step of task analysis. Without any discussion of the design requirements, it is very hard to convince a reader that your model will solve the problem. In particular, you should consider how to make the case that the structure you are visually showing actually benefits the target end user. For example, many authors new to information visualization simply assert, without justification, that showing the hyperlink structure of the web will benefit end users who are searching for information. One of my own early papers fell prey to this very pitfall [26]. However, after a more careful task analysis, I concluded that most searchers do not need to build a mental model of the structure of the search space, so showing them that structure adds cognitive load rather than reduces it. In a later paper [25], I argued that a visual representation of that hyperlink structure could indeed benefit a specific target community, that of webmasters and content creators responsible for a particular site.

The foundation of information visualization is the characterization of how known facts about human perception should guide visual encoding of abstract datasets. The effectiveness of perceptual channels such as spatial position, color, size, shape, and so on depends on whether the data to encode is categorical, ordered, or quantitative [24]. Many individual perceptual channels are preattentively processed in parallel, yet most combinations of these channels must be serially searched [12]. Some perceptual channels are easily separable, but other combinations are not [41, Chapter 5]. These principles, and many others, are a critical part of infovis theory. The last three pitfalls in this section are a few particularly egregious examples of ignoring this body of knowledge.

Hammer In Search Of Nail: If you simply propose a nifty new technique with no discussion of who might ever need it, it's difficult to judge its worth. I am not arguing that all new techniques need to be motivated by specific domain problems: infovis research that begins from a technique-driven starting place can be interesting and stimulating. Moreover, it may be necessary to build an interactive prototype and use it for dataset exploration before it's possible to understand the capabilities of a proposed technique.

However, before you write up the paper about that hammer, I urge you to construct an understanding what kind of nails it can handle. Characterize, at least with some high-level arguments, the kinds of problems where your new technique shines as opposed to those where it performs poorly.

2D Good, 3D Better: The use of 3D rather than 2D for the spatial layout of an abstract dataset requires careful justification that the benefits outweigh the costs [36]. The use of 3D is easy to justify when a meaningful 3D representation is implicit in the dataset, as in airflow over an airplane wing in flow visualization or skeletal structure in medical visualization. The benefit of providing the familiar view is clear, because it matches the mental model of the user. However, when the spatial layout is chosen rather than given, as in the abstract datasets addressed through infovis, there is an explicit choice about which variables to map to spatial position. It is unacceptable, but all too common with naive approaches to infovis, to simply assert that using an extra dimension must be a good idea.

The most serious problem with a 3D layout is occlusion. The ability to interactively change the point of view with navigational controls does not solve the problem. Because of the limitations of human memory, comparing something visible with memories of what was seen before is more difficult than comparing things simultaneously visible side by side [30]. A great deal of work has been devoted to the exploration of the power of multiple linked 2D views, often directly showing derived variables [43] or using them for ordering [2, 23]. In many cases, these views are more effective than simply jumping to 3D [40]. Other difficulties of 3D layouts are that users have difficulty in making precise length judgements because of perspective foreshortening [38], and also that text is difficult to read unless it is billboarded to a 2D plane [9].

Color Cacophony: An infovis paper loses credibility when you make design decisions with blatant disregard for basic color perception facts. Examples include having huge areas of highly saturated color, hoping that color coding will be distinguishable in tiny regions, using more nominal categories than the roughly one dozen that can be distinguishable with color coding, or using a sequential scheme for diverging data. Using a red/green hue coding is justifiable only when strong domain conventions exist, and should usually be redundantly coded with luminance differences to be distinguishable to the 10% of men who are colorblind. You should not attempt to visually encode three variables through the three channels of red, green, and blue; they are not separable because they are integrated by the visual system into a combined percept of color. These principles have been clearly explained by many authors, including Ware [41, Chapter 4].

Rainbows Just Like In The Sky: The unjustified use of a continuous rainbow colormap is a color pitfall so common that I give it a separate title. The most critical problem is that the standard rainbow colormap is perceptually nonlinear. A fixed range of values that are indistinguishable in the green region would clearly show change in other regions such as where orange changes to yellow or cyan changes to blue. Moreover, hue does not have an implicit perceptual ordering, in contrast to other visual attributes such as greyscale or saturation. If the important aspect of the information to be encoded is low-frequency change, then use a colormap that changes from just one hue to another, or has a single hue that changes saturation. If you are showing high-frequency information, where it is important to distinguish and discuss several nameable regions, then a good strategy is to explicitly quantize your data into a segmented rainbow colormap. These ideas are discussed articulately by Rogowitz and Treinish [32].

4 Late Pitfalls: Paper Strategy, Tactics, and Results

The time to consider the late pitfalls is after the bulk of the project work has been done, but before starting to write your paper draft.

4.1 Strategy Pitfalls

Strategy pitfalls pertain to the content and structure of the entire paper, as opposed to more localized tactics problems that only affect one or a few sections.

What I Did Over My Summer Vacation: Do not simply enumerate all activities that required effort when writing a paper. Instead, make judgements about what to discuss in a paper based on your research contributions. This category evokes the grade-school essays many of us were asked to write each fall, which were typically narrative and chronological. Often these Summer Vacation papers contain too much low-level detail, in the extreme cases reading more like a manual than a research paper. For instance, a feature that took weeks or even months of implementation effort may only merit a few sentences, and some features may not be sufficiently interesting to the research community to mention at all. These papers are usually the result of authors who do not know the literature well enough to understand what is and is not a contribution. The solution is to plunge into a more extensive literature review.

Least Publishable Unit: Do not try to squeeze too many papers out of the same project, where you parcel out some tiny increment of research contribution beyond your own previous work. The determination of what is a paper-sized unit of work is admittedly a very individual judgement call, and I will not attempt to define the scope here. As a reviewer, I apply the “I know it when I see it” standard.

Dense As Plutonium: Do not try to cram many papers’ worth of content and contributions into one — the inverse pitfall to the one above. These papers are difficult to read because the barrage of ideas is so dense. More importantly, because you don’t have enough room to explain the full story of how you have accomplished your work, these papers fail the reproducability test. If you are leaving out so many critical details that the work cannot be replicated by a sufficiently advanced graduate student, then split your writeup into multiple papers.

Bad Slice and Dice: If you have done two papers’ worth of work and choose to write two papers, you can still make the wrong choice about how to split up the work between them. In this pitfall, neither paper is truly standalone, yet both repeat too much content found in the other. Repartitioning can solve this problem.

4.2 Tactical Pitfalls

The tactical pitfalls are localized to one or a few sections, as opposed to the paper-level strategy problems above.

Stealth Contributions: Do not leave your contributions implicit or unsaid, whether from intellectual sloth, misplaced modesty, or the hope that the reader may invent a better answer than what you can provide. It is a central piece of your job as an author to clearly and explicitly tell the reader the contributions of your work.

I highly recommend having a sentence near the end of the introduction that starts, "The contribution of this work is", and of using bulleted lists if there are multiple contributions. More subtle ways of stating contributions, using verbs like 'present' and 'propose', can make it more difficult for readers and reviewers to ferret out which of your many sentences is that all-important contributions statement. Also, do not assume that the reader can glean your overall contributions from a close reading of the arguments in your previous work section. While it is critical to have a clear previous work section that states how you address the limitations of the previous work, as I discuss below, your paper should clearly communicate your contributions even if the reader has skipped the entire previous work section.

I find that articulating the contributions requires very careful consideration and is one of the hardest parts of writing up a paper. They are often quite different than the original goals of the project, and often can only be determined in retrospect. What can we do that wasn't possible before? How can we do something better than before? What do we know that was unknown or unclear before? The answers to these questions should guide all aspects of the paper, from the high-level message to the choice of which details are worth discussing. And yet, as an author I find that it's hard to pin these down at the beginning of the writing process. This reason is one of the many to start writing early enough that there is time to refine through multiple drafts. After writing a complete draft, then reading through it critically, I can better refine the contributions spin in a next pass.

I Am So Unique: Do not ignore previous work when writing up your paper. You have to convince the reader that you have done something new, and the only way to do that is to explain how it is different than what has already been done. All research takes place in some kind of intellectual context, and your job as author is to situate what you have done within a framework of that context. A good previous work section is a mini-taxonomy of its own, where you decide on meaningful categorization given your specific topic.

Proposing new names for old techniques or ideas may sneak your work past some reviewers, but will infuriate those who know of that previous work. This tactic will also make you lose credibility with knowledgeable readers. If you cannot find anything related to what you have done, it's more likely that you're looking in the wrong subfield than that your work is a breakthrough of such magnitude that there is no context. Remember to discuss not only work that has been done on similar problems to your own, but also work that uses similar solutions to yours that occurs in different problem domains. This advice is even more critical if you were lax about doing a literature review before you started your project. If you find work similar to your own, you have a fighting chance of carefully differentiating yours in the writeup, but if a reviewer is the one to bring it to your attention, the paper is most likely dead.

Enumeration Without Justification: Simply citing the previous work is necessary but not sufficient. A description that "X did Y", even if it includes detail, is not enough. You must explain why this previous work does not itself solve your problem, and what specific limitations of that previous work your approach does address. Every paper you cite in the previous work section is a fundamental challenge to the very existence of your project. Your job is to convince a skeptical reader that the world needs your new thing because it is somehow better than a particular old thing. Moreover, it's not even enough to just make the case that yours is different – yours must be *better*. The claims you make must, of course, be backed up by your validation in a subsequent results section.

A good way to approach the previous work section is that you want to tell to a story to the reader. Figure out the messages you want to get across to the reader, in what order, and then use the references to help you tell this story. It is possible to group the previous work into categories, and to usefully discuss the limitations of the entire category.

Sweeping Assertions: A research paper should not contain sweeping unattributed assertions. You have three choices: cite your source; delete the assertion from your paper; or explicitly tag the statement as your observation, your conjecture, or an explanation of your results. In the last case, the assertion is clearly marked as being part of your research contribution. Be careful with folk wisdom that "everybody knows". You could be mistaken, and tracking down the original sources may change or refine your views. If you cannot find a suitable source after extensive digging, you have stumbled upon a great topic for a future paper! You may either validate and extend the conventional wisdom, or show that it is incorrect.

I Am Utterly Perfect: No work is perfect. An explicit discussion of the limitations of your work strengthens, rather than weakens, your paper. Papers without a discussion of limitations, weaknesses, and implications feel unfinished or preliminary. For instance, how large of a dataset can your system handle? Can you categorize the kinds of datasets for which your technique is suitable and those for which it is not?

4.3 Results Pitfalls

Several pitfalls on how to validate your claims can occur in the results section of your paper.

Unfettered By Time: Do not omit time performance from your writeup, because it is almost always interesting and worth documenting. The level of detail at which you should report this result depends on the paper type and the contribution claims. For instance, a very high-level statement like "interactive response for all datasets shown on a desktop PC" may suffice for an evaluation paper or a design study paper. However, for a technique paper with a contribution claim of better performance than previous techniques, detailed comparison timings in tables or charts would be a better choice.

Fear and Loathing of Complexity: Although most infovis papers do not have detailed proofs of complexity, technique papers that focus on accelerating performance should usually include some statement of algorithm complexity.

Straw Man Comparison: When comparing your technique to previous work, compare against state-of-the-art approaches rather than outdated work. For example, authors unaware of recent work in multilevel approaches to force-directed graph drawing [10] sometimes compare against very naive implementations of spring systems. At the lower level, if you compare benchmarks of your implementation to performance figures quoted from a previous publication and your hardware configuration is more powerful, you should explicitly discuss the difference in capabilities. Better yet, rerun the benchmarks for the competing algorithms on the same machine you use to test your own.

Tiny Toy Datasets: Avoid using only tiny toy datasets in technique papers that refine previously proposed visual encodings. While small synthetic benchmarks can be useful for expository purposes, your validation should include datasets of the same size used by state-of-the-art approaches. Similarly, you should use datasets characteristic of those for your target application.

On the other hand, relatively small datasets may well be appropriate for a user study, if they are carefully chosen in conjunction with some specific target task and this choice is explained and justified.

But My Friends Liked It: Positive informal evaluation of a new infovis system by a few of your infovis-expert labmates is not very compelling evidence that a new technique is useful for novices or scientists in other domains. While the guerilla/discount methodology is great for finding usability problems with products [27], a stronger approach would be informal evaluation with more representative subjects, or formal evaluation with rigorous methodology.

Unjustified Tasks: Beware of running a user study where the tasks are not justified. A study is not very interesting if it shows a nice result for a task that nobody will ever actually do, or a task much less common or important than some other task. You need to convince the reader that your tasks are a reasonable abstraction of the real-world tasks done by your target users. If you are the designer of one of the systems studied, be particularly careful to make a convincing case that you did not cherry-pick tasks with a bias to the strengths of your own system.

5 Final Pitfalls: Style and Submission

After you have a full paper draft, you should check for the final-stage pitfalls.

5.1 Writing Style Pitfalls

Several lower-level pitfalls pertain to writing style.

Deadly Detail Dump: When writing a paper, do not simply dump out all the details and declare victory. The details are the *how* of what you did and do belong at the heart of your paper. But you must first say *what* you did and *why* you did it before the *how*. This advice holds at multiple levels. For the high-level paper structure, start with motivation: why should I, the reader, care about what you've done? Then provide an overview: a big-picture view of what you did. The algorithmic details can then appear after the stage has been set. At the section, subsection, and sometimes even paragraph level, stating the *what* before the *how* will make your writing more clear.

Story-Free Captions: Avoid using a single brusque sentence fragment as your caption text. Caption words are not a precious resource that you should hoard and spend begrudgingly. Instead, design your paper so that as much of the paper story as possible is understandable to somebody who flips through looking only at the figures and captions. Many readers of visualization and graphics papers do exactly this when skimming, so make your captions as standalone as possible.

My Picture Speaks For Itself: You should talk the reader through how your visual representation exposes meaningful structure in the dataset, rather than simply assuming the superiority of your method is obvious to all readers from unassisted inspection of your result images. Technique and design study papers usually include images in the results section showing the visual encodings created by a technique or system on example datasets. The best way to carry out this qualitative evaluation is to compare your method side-by-side with representations created by competing methods on the same dataset.

Grammar is Optional: Grammar is not optional; you should use correct syntax and punctuation for a smooth low-level flow of words. If English is not your first language, consider having a native speaker check the writing before submitting a paper for review, and also before the final version of your paper goes to press. I recommend Dupré's book [4] as an excellent pitfall-oriented technical writing guide for computer scientists.

Mistakes Were Made: Avoid the passive voice as much as possible. I call out this particular grammar issue because it directly pertains to making your research contribution clear. Is the thing under discussion part of your research contribution, or something that was done or suggested by others? The problem with the passive voice is its ambiguity: the reader does not have enough information to determine *who* did something. This very ambiguity can be the lure of the passive voice to a slothful or overly modest writer. I urge you to use the active voice and make such distinctions explicitly.

Jargon Attack: Avoid jargon as much as possible, and if you must use it then define it first. Definitions are critical both for unfamiliar terms or acronyms, as well as for standard English words being used in a specific technical sense.

Nonspecific Use Of Large: Never just use the words 'large' or 'huge' to describe a dataset or the scalability of a technique without giving numbers to clarify the order of magnitude under discussion: hundreds, tens of thousands, millions? Every author has a different idea of what these words mean, ranging from 128 to billions, so be specific. Also, you should provide the size of all datasets used in results figures, so that readers don't have to count dots in an image to guess the numbers.

5.2 Submission Pitfalls

Finally, I caution against pitfalls at the very end of the project, when submitting your paper.

Slimy Simultaneous Submission: Simultaneous submission of the same work at multiple venues who clearly request original work is highly unethical. Moreover, simultaneous submission is stupid, because it is often detected when the same reviewer is independently selected by different conference chairs. The number of experts in any particular subfield can be quite a small set. The standard penalty upon detection is instant dual rejection, and multi-conference blacklists are beginning to be compiled. Finally, even if you do succeed in getting the same work published twice, any gains you make by having a higher publication count will be offset when you lose credibility within your field from those who actually read the work and are annoyed to wade through multiple papers that say the same thing.

Resubmit Unchanged: If your paper is rejected, don't completely ignore the reviews and resubmit to another venue without making any changes. As above, there's a reasonable chance that you'll get the one of the same reviewers again. That reviewer will be highly irritated.

6 Pitfalls By Generality

A cross-cutting way to categorize these pitfalls is by generality. Many hold true for any scientific research paper, rather than being specific to visualization. Of the latter, many hold true for both scientific visualization (scivis) and information visualization (infovis). As many have lamented, the names of these subfields are unfortunate and confusing for outsiders. The definition I use is that it's infovis when the spatial representation is chosen, and it's scivis when the spatial representation is given. Operationally, InfoVis split off as a sister conference from IEEE Visualization (Vis) in 1995. At Vis, the focus is now on scivis.

The choice of paper types is specific to the InfoVis author guide, because this categorization is not explicitly discussed in the Vis call for papers. The first-stage type pitfalls are thus quite specific to infovis. The middle pitfalls on visual encoding are specific to visualization. *Color Cacophony* and *Rainbows Just Like In The Sky* certainly pertain to both infovis and scivis. *Unjustified Visual Encoding*, *Hammer In Search Of Nail*, and *2D Good, 3D Better* focus on issues that are more central for an infovis audience, but may well be of benefit to scivis as well. All of the strategy pitfalls pertain to any research paper. The result pitfalls hold for all visualization papers, and *Straw Man Comparison* is general enough for all research papers. The tactical and final stage pitfalls are very general, with two exceptions. *Story-Free Captions* is specific to both visualization and computer graphics. *My Picture Speaks For Itself* is again most tuned for infovis, but certainly may pique the interest of the scivis community.

Although I have framed my discussion in terms of the InfoVis author guide paper categories, my comments also apply to infovis papers in other venues. I argue that even if a call for papers does not explicitly state paper categories, keeping this paper taxonomy in mind will help you write a stronger paper.

7 Conclusion

I have advocated an approach to conducting infovis research that begins with an explicit consideration of paper types. I have exhorted authors to avoid pitfalls at several stages of research process, including visual encoding during design, a checkpoint before starting to write, and after a full paper draft exists. My description and categorization of these pitfalls reflects my own experiences as author, reviewer, and papers chair. I offer it in hopes of steering and stimulating discussion in our field.

8 Acknowledgements

This paper has grown in part from a series of talks. The impetus to begin articulating my thoughts was the Publishing Your Visualization Research panel at the IEEE Visualization 2006 Doctoral Colloquium. I benefited from discussion with many participants at the 2007 Dagstuhl Seminar on Information Visualization and the 2007 Visual Interactive Effective Worlds workshop at the Lorentz Center, after speaking on the same topic. I also appreciate a discussion with Pat Hanrahan and Lyn Bartram on design studies. John Stasko's thoughts considerably strengthened my discussion of model papers. I thank Aaron Barsky, Hamish Carr, Jeff Heer, Stephen Ingram, Ciarán Llachlan Leavitt, Peter McLachlan, James Slack, and Matt Ward for feedback on paper drafts. I particularly thank Heidi Lam, Torsten Möller, John Stasko, and Melanie Tory for extensive discussions beyond the call of duty.

References

1. R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 111–117, 2005.
2. R. A. Becker, W. S. Cleveland, and M. J. Shyu. The visual design and control of trellis display. *Journal of Computational and Statistical Graphics*, 5:123–155, 1996.
3. S. Card and J. Mackinlay. The structure of the information visualization design space. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 92–99, 1997.
4. L. Dupré. *Bugs in Writing*. Addison Wesley, 1995.
5. J.-D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, 2002.
6. Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for visualizing large multivariate data sets. In *Proc. IEEE Visualization Conf. (Vis)*, 1999.
7. G. Furnas and B. Bederson. Space-scale diagrams: Understanding multiscale interfaces. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 1995.
8. G. W. Furnas. A fisheye follow-up: Further reflection on focus + context. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 999–1008, 2006.
9. T. Grossman, D. Wigdor, and R. Balakrishnan. Exploring and reducing the effects of orientation on text readability in volumetric displays. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 483–492, 2007.
10. S. Hachul and M. Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In *Proc. Graph Drawing (GD)*, volume 3383 of *LNCS*, pages 285–295. Springer-Verlag, 2004.
11. S. Havre, B. Hetzler, and L. Nowell. Themeriver(tm): In search of trends, patterns, and relationships. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 115–123, 2000.
12. C. G. Healey. Perception in visualization. `http://www.csc.ncsu.edu/faculty/healey/PP`, cited 14 Nov 2007.
13. J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 421–430, 2005.
14. J. Heer and danah boyd. Vizster: Visualizing online social networks. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 32–39, 2005.
15. D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis 06)*, 12(5):741–748, Sept. 2006.
16. K. Hornbæk and M. Hertzum. Untangling the usability of fisheye menus. *ACM Trans. on Computer-Human Interaction (ToCHI)*, 14(2), 2007.
17. E. Isaacs and J. Tang. Why don't more non-North-American papers get accepted to CHI? *SIGCHI Bulletin*, 28(1), Jan. 1996. `http://www.sigchi.org/bulletin/1996.1/isaacs.html`.
18. C. Johnson, R. Moorhead, T. Munzner, H. Pfister, P. Rheingans, and T. S. Yoo. *NIH/NSF Visualization Research Challenges Report*. IEEE Computer Society Press, 2006.
19. R. E. Johnson et al. How to get a paper accepted at OOPSLA. In *Proc. Conf. Object Oriented Programming Systems Languages and Applications (OOPSLA)*, pages 429–436, 1996. `http://www.sigplan.org/oopsla/oopsla96/how93.html`.

20. J. Kajiya. How to get your SIGGRAPH paper rejected. `http://www.siggraph.org/publications/instructions/rejected`.
21. R. Kincaid, A. Ben-Dor, and Z. Yakhini. Exploratory visualization of array-based comparative genomic hybridization. *Information Visualization*, 4(3):176–190, 2005.
22. R. Levin and D. D. Redell. An evaluation of the ninth SOSP submissions; or, how (and how not) to write a good systems paper. *ACM SIGOPS Operating Systems Review*, 17(3):35–40, July 1983. `http://www.usenix.org/events/samples/submit/advice.html`.
23. A. MacEachren, X. Dai, F. Hardisty, D. Guo, and G. Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 31–38, 2003.
24. J. D. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. on Graphics (TOG)*, 5(2):111–141, 1986.
25. T. Munzner. Drawing large graphs with H3Viewer and Site Manager. In *Proc. Graph Drawing (GD)*, volume 1547 of *LNCS*, pages 384–393. Springer-Verlag, 1998.
26. T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3D hyperbolic space. In *Proc. Virtual Reality Modelling Language Symposium (VRML)*, pages 33–38. ACM SIGGRAPH, 1995.
27. J. Nielsen. Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. In R. G. Bias and D. J. Mayhew, editors, *Cost-justifying usability*, pages 245–272. Academic Press, 1994.
28. C. Partridge. How to increase the chances your paper is accepted at ACM SIGCOMM. *ACM SIGCOMM Computer Communication Review*, 28(3), July 1998. `http://www.sigcomm.org/conference-misc/author-guide.html`.
29. C. Plaisant. The challenge of information visualization evaluation. In *Proc. Advanced Visual Interfaces (AVI)*, pages 109–116, 2004.
30. M. Plumlee and C. Ware. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *Proc. ACM Trans. on Computer-Human Interaction (ToCHI)*, 13(2):179–209, 2006.
31. Z. Pousman, J. T. Stasko, and M. Mateas. Casual information visualization: Depictions of data in everyday life. *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis 07)*, 13(6):1145–1152, 2007.
32. B. E. Rogowitz and L. A. Treinish. How not to lie with visualization. *Computers In Physics*, 10(3):268–273, May/June 1996. `http://www.research.ibm.com/dx/proceedings/pravda/truevis.htm`.
33. M. Shaw. Mini-tutorial: Writing good software engineering research papers. In *Proc. Intl. Conf. on Software Engineering (ICSE)*, pages 726–736, 2003. `http://www.cs.cmu.edu/~Compose/shaw-icse03.pdf`.
34. J. Shewchuk. Three sins of authors in computer science and math. `http://www.cs.cmu.edu/~jrs/sins.html`, 1997.
35. B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*, pages 38–43, 2006.
36. H. S. Smallman, M. S. John, H. M. Oonk, and M. B. Cowen. Information availability in 2D and 3D displays. *IEEE Computer Graphics and Applications (CG&A)*, 21(5):51–57, 2001.
37. D. Tang, C. Stolte, and R. Bosch. Design choices when architecting visualizations. *Information Visualization*, 3(2):65–79, June 2004.

38. M. Tory, A. E. Kirkpatrick, M. S. Atkins, and T. Möller. Visualization task performance with 2D, 3D, and combination displays. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 12(1):2–13, 2006.
39. J. G. Trafton, S. S. Kirschenbaum, T. L. Tsui, R. T. Miyamoto, J. A. Ballas, and P. D. Raymond. Turning pictures into numbers: Extracting and generating information from complex visualizations. *Intl. Journ. Human Computer Studies*, 53(5):827–850, 2000.
40. J. J. van Wijk and E. R. van Selow. Cluster and calendar based visualization of time series data. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 4–9, 1999.
41. C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann/Academic Press, 2nd edition, 2004.
42. C. Weaver, D. Fyfe, A. Robinson, D. W. Holdsworth, D. J. Peuquet, and A. M. MacEachren. Visual analysis of historic hotel visitation patterns. *Information Visualization*, 6(1):89–103, Feb 2007.
43. L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, 2005.
44. B. Yost and C. North. The perceptual scalability of visualization. *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis 06)*, 12(5):837–844, Sept. 2006.