

Characterising the Quality of Online Task-Based Application Tutorials

by

Matthew Graham Lount

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
July 2013

© Copyright 2013 by Matthew Graham Lount

Thesis advisor

Andrea Bunt

Author

Matthew Graham Lount

Characterising the Quality of Online Task-Based Application Tutorials

Abstract

End-user authored tutorials are increasingly becoming the norm for assisting users with learning software applications, but little is known about the quality of these tutorials. Using metrics derived from previous work, I characterize the quality of text- and image-based Photoshop tutorials available to users online. I compare these tutorials across four sources representing tutorials that are, i) written by a close-knit online community, ii) written by expert users, iii) most likely to be found, and iv) representative of the general population of tutorials. I found that not only are expert users generally writing higher quality tutorials than the other authors, but also, many of the typical tutorials are suffering from some important limitations. Most notably, they often lack attempts to help users avoid common errors, and seldom provide users with appropriate amounts of reasoning for undertaking steps. I also examine a typical tutorial rating system and find that it does not sufficiently distinguish quality between tutorials. I demonstrate the use of my findings by presenting two applications that I designed: a tutorial authoring tool, and a tutorial presentation site.

Contents

Abstract	ii
Table of Contents	iv
List of Figures	v
List of Tables	vi
1 Introduction	1
2 Related Work	5
2.1 Designing Effective Instructional Materials	5
2.1.1 Differences Between Media	6
2.1.2 Designing Instructions	6
2.2 Supporting Tutorial Authors and Consumers	7
2.2.1 Automating Tutorial Creation	7
2.2.2 Simplifying Tutorial Search	8
2.2.3 Reinventing Tutorials	9
Modifying Existing Tutorials	9
Unique Learning Documents	10
2.3 Support for Users of Complex Software	12
2.3.1 Interface Customization	12
2.3.2 In-Application Assistance	14
2.4 Ratings	15
2.5 Summary	17
3 Measuring Quality	18
3.1 Showcasing Metrics	19
3.2 Metrics from Related Work	23
3.3 Metrics from Users	26
3.4 Summary	26
4 A Study of Online Photoshop Tutorial Quality	27
4.1 Tutorial Sources	28

4.1.1	Application-Centered Community	28
4.1.2	Tutorial Aggregator	29
4.1.3	Tutorial Factory	30
4.1.4	Popular Task-Based Tutorials	31
4.2	Tutorial Sampling	31
4.2.1	Selecting Tutorials for Showcasing Metric Analysis	31
4.2.2	Selecting Tutorials for Quality Metric Analysis	32
	CUTS	32
4.3	Data Collection Method	34
4.3.1	Showcasing Value Collection	34
4.3.2	Metric Value Collection	35
4.4	Results	38
4.4.1	Current Showcasing Strategies	38
4.4.2	Tutorial Quality	41
	Quality Across All Tutorials	42
	Quality Differences Between Communities	47
4.5	Discussion	49
4.5.1	Showcasing Strategies	49
4.5.2	Quality Across All Tutorials	50
4.5.3	Quality Differences Between Communities	51
4.6	Limitations	52
4.6.1	Generalization	53
4.6.2	Missing Factors	54
4.6.3	No User Study	55
4.7	Summary	55
5	Implementation Examples	57
5.1	Tutorial Creator	57
5.2	Tutorial Website	63
5.3	Alternative Implementations	68
5.4	Summary	70
6	Conclusion	71
6.1	Summary	71
6.2	Future Work	73
	Bibliography	76

List of Figures

3.1	Example Showcasing Metrics	21
4.1	Ratings	39
4.2	Vote Boxplots	40
5.1	Tutorial Creator Application Overview	58
5.2	Editing the Title Tile in the Tutorial Creator Application	60
5.3	Editing a Step in the Tutorial Creator Application	61
5.4	Inserting Photoshop Tool References with the Tutorial Creator Application	62
5.5	Selecting a Tutorial in the Tutorial Website	64
5.6	Viewing a Tutorial in the Tutorial Website	67

List of Tables

3.1	Showcasing Metrics	19
3.2	Metrics	24
4.1	Votes by Ratings	41
4.2	Summary of Overall Quality	42
4.3	Analysis of Quantitative Metrics	45
4.4	Analysis of Categorical Metrics	48

Chapter 1

Introduction

In an attempt to meet the increasingly diverse needs of growing user-bases, developers are expanding the abilities of their applications. This increase in capabilities comes with a jump in the number of commands and menus, growing complexity, and an ever-widening barrier to mastery [24, 45]. While the applications have grown, the dominant form of documentation has also changed. Thick, paper manuals explaining what each command does gave way to electronic versions, included on the same CD as the applications. These electronic manuals were soon combined with (and in many cases, replaced by) websites that allow documentation to be updated without having to disseminate the changes directly to the users. In recent years, the dominance of user-authored content on the web (commonly known as ‘Web 2.0’) has resulted in users authoring rich tutorials on how to complete tasks with popular software. With many typical users seeking task-specific help through the use of search engines [12] (e.g., Google), users are now frequently being pointed to the tutorials authored by users rather than the traditional professionally-authored documentation.

This movement from in-application documentation and professionally authored manuals to end-user generated tutorials raises a number of questions about their utility as a help resource; the questions that I focus my thesis on are:

1. What is the quality of the tutorials that are available for users on the web?
2. Does the quality of tutorials vary according to characteristics of the authoring community?
3. Are currently used mechanisms for highlighting tutorial quality (e.g., five-star rating schemes, and the number of views) effectively distinguishing between the range of tutorials available?

I also investigate how the answers to these questions can be used to inform the design of future software systems that support both, the consumers of tutorials, and their authors.

To answer these questions, I collected a series of quality metrics from the literature on effective tutorial design, an informal survey of strategies used by websites to showcase tutorials, and comments left by users who have read the tutorials. Examples of the 30 metrics I assembled include the number of images used in the tutorial, the mean rating assigned to a tutorial by its readers, and whether or not the tutorial included all the source files needed to follow along. I then applied these metrics to 154 text- and image-based Photoshop tutorials sampled across four different tutorial sources: i) a close-knit online authoring community; ii) a site posting only tutorials written by expert users; iii) popular tutorials retrieved through Google Search; and iv) a set of tutorials representative of the general population of tutorials.

I focused my exploration on text- and image-based tutorials for Photoshop, an image-manipulation application with one of the largest showings of online tutorials (e.g., over 18,000 available on www.tutorialized.com). While video tutorials are becoming increasingly popular, there is less literature on what makes for an effective video tutorial than a more conventional text- and image-based tutorial, and their suitability for learning is still debated [23, 50]. As such, I felt that the conventional tutorials represented a good starting point for exploration in this space.

My analysis shows that most tutorials in my sample adhere to many of the guidelines established in the literature, providing initial evidence legitimizing these resources as forms capable of teaching application use. At the same time, there are a number of ways in which the tutorials are falling short, such as providing little assistance for potential difficulties with their instructions, and often failing to draw the user's attention to relevant aspects of the tutorial images. I also found a number of clear differences in the tutorial sources, with the tutorial source contributed to by expert Photoshop users having a number of advantages over the remaining sources. In terms of showcasing mechanisms, I found that five-star ratings tend to cluster around the middle of the scale regardless of tutorial quality, limiting their effectiveness as a discriminator in this particular domain. I used these findings to inform the design of two applications, one for authoring tutorials, and one for showcasing them.

The balance of my thesis is broken up as follows. In chapter 2 I explore works by other authors that informed the body of my thesis, focusing on four key areas: designing effective instructional materials for teaching the use of computer applications; methods for supporting the communities of tutorial authors and consumers;

systems for supporting the use of complex software; and methods for using ratings. In chapter 3, I describe the quality metrics that I use throughout my thesis. Chapter 4 consists of the primary study that I conducted for analyzing the quality of tutorials. In chapter 5, I present two designs for applications that take advantage of the findings of my study. I conclude in chapter 6 by summarizing my findings and discussing potential avenues for future work.

Chapter 2

Related Work

I focus my coverage of related work in four main areas. I first introduce prior work on creating effective instructional materials, revisiting this work in more detail in section 3.2 when I define the quality metrics used in my analysis. I next explore research on systems that aid in tutorial creation and consumption, moving on to works that support users of complex software in other ways than through tutorials. Finally, I discuss existing work on ratings. These works contribute to my analysis of the methods that websites use for showcasing tutorials, and inform the example applications that I designed to demonstrate the uses of my findings.

2.1 Designing Effective Instructional Materials

There is a large body of work on how to best instruct novice users in learning systems. This work began primarily with determining the best media to use, with a focus more recently on how to design the instructions within those chosen media.

2.1.1 Differences Between Media

Early work on designing effective instruction focused on determining the effects that different media had on instruction. For example, Booher [4] and Stern [58] both advocated combining text and images in creating effective instructional materials, with Stern also concluding that voice was particularly bad at disseminating procedural instructions. Later, Palmiter and Elkerton compared video instruction with more traditional text- and image-based tutorials [50, 51]. They found that video has the potential to speed task completion, but only when there is a close match between the task and the video. They also noted that traditional tutorials seem to produce better long-term learning, although, more recently, Grossman and Fitzmaurice [23] found evidence to the contrary in a study of a system that I will describe in section 2.3.2, below. This inconsistent evidence on the benefits and drawbacks of video is part of the reason that I chose to only include traditional text- and image-based tutorials in my study.

2.1.2 Designing Instructions

With a better understanding of how medium choice affects instruction, more recent work has focused on a fine-grained examination of how to design instructions. Of particular note, Carroll introduced a “minimalist approach” to documentation, which advocated for task-centered documentation combined with user-directed exploration rather than the dominant help systems of the time, software manuals [7]. This approach has many benefits, but, according to a review by Farkas and Williams [14], only captures some of the important aspects of documentation. Novick and Morse [48]

explore the role that steps and images play in instruction by studying the use of different styles of tutorials for creating origami objects. Agrawala et al. [2] also studied object assembly instructions, focusing on furniture rather than origami. Both papers present a series of principles of instruction, some of which, such as those focused on how to orient an object, lose relevance in the transition from the three-dimensional world of assembling objects to the two-dimensional world of modifying photographs. Others, such as making use of ‘action’ diagrams rather than ‘structural’ diagrams, resonate strongly in the domain of Photoshop tutorials.

While these works have focused on how different aspects of instruction affect users, my work applies these theoretical aspects as metrics to existing online tutorials in order to facilitate a discussion of the state of tutorials currently available to users.

2.2 Supporting Tutorial Authors and Consumers

Recently, there has been a surge in interest in creating systems that support tutorial authors and consumers. These systems automate tutorial creation, simplify users’ decision-making processes when they are choosing between tutorials, and create alternative forms of tutorials.

2.2.1 Automating Tutorial Creation

A number of researchers have attempted to leverage application logging to automate the creation of tutorials. For instance, Grabler et al. [21] created a system that automatically authors image- and text-based tutorials by combining image-recognition software with command usage logs collected from an initial demonstration

by the author. Chi et al. [8] also use demonstration to automatically generate tutorials, but mix short videos of the users' actions into each of the otherwise textual steps. Videos are kept short in order to avoid the primary pitfalls of video tutorials that have been emphasized in the past; users do not have to keep pausing or scrubbing back in the video to keep up with the tutorial.

2.2.2 Simplifying Tutorial Search

Given the sheer volume of web-based tutorials available, it is not surprising that work has also focused on how to assist users in deciding which tutorials will best support their goals. For example, the help system created by Ekstrand et al. [12] aids users in choosing between tutorials by integrating application context into results from Google, allowing the system to display information about what interface components are referenced in the search results. In a similar fashion, the Delta system displays the commands present, the number of steps, and the product of the tutorial (in this case a final image) for groups of tutorials [30]. Both of these works highlight the commands that are used in the system under the premise that users want tutorials to use commands that they are familiar with. This, however, can have negative impacts on the amount of learning that a user experiences within the tutorial, as exposure to unfamiliar commands is a key aspect of learning to use an application [7]. Work on explicitly increasing command exposure will be discussed in section 2.3.2.

2.2.3 Reinventing Tutorials

Many researchers have created alternative forms of tutorials. Some researchers have done so by modifying existing tutorials to make them more effective, while others have created systems that enable users to author unique learning documents.

Modifying Existing Tutorials

Some researchers have sought to improve the large body of existing tutorials. For example, Pongnumkul et al. [54] sought to overcome problems that users have with following video tutorials. To this end, they created an application, Pause-and-Play, that uses computer-vision techniques to automatically pause and play video tutorials at appropriate times as users work along with the tutorial. They found that in some cases, this approach can prevent the user from exploring outside of the tutorial, or from easily correcting errors from previous steps. Overall, though, they found that users prefer working with this application to working with video tutorials on their own.

Laput et al. [35] also augmented preexisting tutorials, but focused on traditional text and image tutorials, introducing what they call tutorial-based applications, or *tapps* for short, and an application called TappCloud. A tapp is essentially a combination of a tutorial and a series of macros that complete that tutorial; users can adjust parameters for any step, and TappCloud allows them to view the results of subsequent steps performed using their own resources. Tapps are created semi-automatically from existing tutorials using a combination of text-analysis (based on work by Fourney et al. [17]) and machine learning heuristics (using research by Lau et al. [36]). When

the system fails to appropriately extract a step from a tutorial, it relies on a demonstration by the tapp author (using the work of Grabler et al. [21], already described above in section 2.2.1). This work blends the application and the tutorial together, making each tutorial into an application that functions on users' own images, and is usable from a web page. The benefit of this type of system is that users can easily and accurately execute the steps of a tutorial. However, it is possible for the user to execute the tapp with a single button click, without ever viewing the steps of the tutorial; in this case, the skills demonstrated in the tutorial would not be learned by the user.

Unique Learning Documents

Several researchers have developed systems that create documents that combine tutorials with unique interactions. These interactions help the user accurately follow along with the tutorial, and enable them to more completely explore the design process.

For instance, Kelleher and Pausch [28] created a tool for authoring an alternative form of tutorial, which they call a *stencil*. The stencils that are created integrate into the target application, and prevent users from making mistakes while completing the tutorial by putting up a shield-like translucent overlay over the application's interface. The overlay captures keyboard and mouse clicks, and only passes them to the actual interface if the user's action targets the correct function for advancing through the stencil. This allows tutorial creators to dictate exactly what steps need to be taken in order to complete the tutorial, and assures that users will not skip steps.

Grossman et al. [25] used application logging to create a document-exploration tool, Chronicle. This application allows users to record and view videos of the creation of image files. It provides several interactions to locate points of interest within the history, such as tools to explore the life-cycle of specific areas, and annotated timelines that highlight command usage. Denning et al. [10] created a similar application, MeshFlow, which presents users with visualizations of the construction sequence of a three dimensional mesh model from its logged command history.

Lafreniere et al. [34] created an online system that records tutorial authors using a built-in image editing application, and integrates these videos into step-based tutorials. The system then records video of any user completing the tutorial, and makes those videos available for users to view along with the author's original videos. This system allows users to see the same tutorial step completed in different contexts, potentially enabling them to view a video that matches more closely to their own task than the original author's.

The works in this section have attempted to support tutorial authors and consumers through the modification of existing tutorials, and the creation of new styles of help documents. An important step which has been ignored up until now, however, is identifying areas in which current tutorials are actually lacking, highlighting where efforts in supporting tutorial communities are most needed. My work will support any efforts made in this area in the future by identifying those problem areas.

2.3 Support for Users of Complex Software

Tutorials are only one technique that has been employed to help users with complex software. Other researchers have focused on alternatives, including reducing complexity through interface customization, and integrating context-aware assistance into the application.

2.3.1 Interface Customization

A very popular technique that has been studied for aiding users with the growing complexity of software is the customization of user interfaces. This customization can be adaptive (completed entirely by the system), adaptable (customized by the user), or mixed (the user approves customizations suggested by the system).

Adaptive interface customization centers on menu and toolbar item reorganization (e.g., [20, 22, 52, 56]), reorganizing based on frequency or recency of a menu item's use. Despite the theoretical gains from this reordering, Gajos et al. [19] explained that in practice these menus have had mixed reactions from users who often find the reorganization disorienting. In evaluating past adaptive menus, they concluded that split menus where commands are copied from their original location to the top section of menus (rather than moved, as Sears et al. [56] originally implemented), resulted in the highest perceived benefit and lowest perceived cost of the existing adaptive menus.

Adaptable interfaces also focus on reorganizing toolbar and menu items (e.g., [59]), although the user is in charge of assigning the order, thus reducing the disorientation associated with adaptive menus. Many researchers have conducted tests between

adaptable menus, and the different forms of adaptive menus that have been invented (e.g., [16, 44, 46, 52]), generally finding that adaptable menus are more effective than adaptive.

Adaptable interfaces are not without their challenges, however, as in practice, users frequently do not customize their interface [49], nor recustomize it when their usage patterns change [44], losing out on many potential benefits. AdaptableGIMP [32, 40, 41] is an application that partially addresses this concern. It makes customization a primary feature of the interface, allowing users to dynamically choose a set of components for the main interface based on the task that they are performing. These sets of tools and commands, called *task sets*, are created by the community of users of the application, and are made available to search through and install instantly from within the application, offsetting the concern that many users will not customize because of the time it takes. Users are further aided by the ability for the community to write documentation or tutorials for the task sets, so users should be able to use the task set after they've installed it.

An intuitive answer to the criticisms of adaptable and adaptive customization was to attempt to offset the negatives of each by combining them. With a mixed-initiative user interface, the system attempts to determine improvements for the user interface while the user uses the system, but waits for some action from the user to implement these changes. For example, Bunt et al. [6] implemented MICA, which makes interface customization suggestions for Microsoft Word, providing a quantitative prediction of the time customizing will save.

2.3.2 In-Application Assistance

Some researchers have investigated forms of application-integrated context-aware assistance as a method for helping users master complex applications.

A well known, early example of this is Horvitz et al. [27], who developed a set of processes for constructing user models from Bayesian networks, allowing them to predict a user's temporal goals and needs. They then developed the Lumiere System, a computerized assistant that provides contextual assistance when it thinks it knows enough about what the user is doing, and that they would like help doing it.

In order to enhance a user's breadth of knowledge of commands, which Carroll [7] found to be integral to system mastery, Matejka et al. [43] integrate Community-Commands, a command-suggesting recommender system, into Autodesk's AutoCAD application. These suggestions are generated by comparing a user's command-use history with that of other users, and presenting commands that are expected to be both unfamiliar and useful. Command recommendation has also been proposed previously by Linton et al. [38], although the system they described was intended to work on a smaller scale, primarily for propagating knowledge of commands between users in a single workspace or organization.

Another form of in-application assistance comes in the form of tool tips [13], small textual annotations that appear when a user hovers the mouse over an object of interest. While they have been found to be successful at aiding users in their search for functionality [11], they are not as good at teaching users to use tools [13]. Grossman and Fitzmaurice [23] modified these tool tips to create ToolClips, annotations that appear as tool tips do, but provide users with short video demonstrations of how to

use the tools.

Matejka et al. [42] also addressed the issue of teaching the functionality of an application. In this case, the system automatically loads a set of videos and textual help onto a second display while the user uses the system. This system helps users by performing the search for information before the user knows they need it, preventing them from having to switch contexts when they do need it.

These papers have all covered ways of supporting end-users with their use of complex applications in an unguided way. Tutorials serve a similar purpose but function to aid end-users with tasks for which they are explicitly looking for assistance.

2.4 Ratings

Research on ratings in computer science has focused primarily on how they pertain to recommender systems. My work looks at ratings as they pertain to determining quality of something rather than determining whether a specific user will like it; there are a number of rating-related issues that are common between these tasks. These works strongly impact my analysis of the ratings in chapter 4, and I use many of the findings from these papers to inform the designs of the rating scales in the application prototypes that I present in chapter 5.

How to present ratings to users is of great interest to the scientific community. Cosely et al. [9], for instance, used the popular movie-recommender site, Movie Lens, to conduct several studies on the properties of rating scales. One finding of interest to me in designing my prototypes was that users prefer ratings with more granularity. Another relevant finding was that users can be manipulated into rating something

higher by first viewing a higher rating. This finding is strengthened by a classic finding in the literature on conformity, where Asch [3] shows that people are easily swayed by other peoples' opinions. Nass and Moon's [47] finding that people react similarly to computers as they do to people can help lend credibility to the idea that people may view the ratings in a similar way to directly being privy to a person's opinions.

Sparling and Sen [57] also studied different rating scales, particularly trying to determine the difference in costs to the user associated with them. They had participants rate movies, and the helpfulness of product reviews, using four popular interfaces: a unary 'like' button, binary 'thumbs up/down' buttons, a five-star scale, and a 100-point slider. The authors found that users rate on the extreme end points of a scale faster than in the middle, and that users are more likely to rate multiple values when the scale is less fine-grained. Later, I use these findings to help justify my choice of rating scales used in my implementation examples.

Many works have focused on finding algorithms that use ratings to generate good recommendations for users through what is known as collaborative filtering (e.g., [5, 15, 31, 39, 53, 61]). Recently, some researchers have attempted to answer whether collecting ratings on multiple scales can provide better predictions than the classically-used single-rating systems. For example, Sahoo et al. [55], and Adomacivius and Kwon [1] each developed two algorithms (for a total of four) that generate movie recommendations using multiple category ratings collected from a popular movie website, Yahoo! Movies. In all of these algorithms, a single value for each movie is generated from the multiple categories in order to rank them. By contrast, Lee and

Teng [37] reject the idea of combining ratings of multiple scales into a single value, indicating that this results in a loss of information that users could find useful. They propose a method for returning results that takes advantage of the fact that some users will put different weights on different properties of the items. These studies suggest that there is value in breaking up how users rate things into multiple categories.

2.5 Summary

Based on the above literature review, we see that there is a great deal of interest in the research community for supporting tutorial creation and use, but little is actually known about the current state of online tutorials, and where that support is best targeted. There have also been a lot of investigations into how to properly instruct users to facilitate learning and task-completion, but to my knowledge no explicit attempts have been made to measure how well web-based tutorials comply with the lessons learned from these investigations. We also saw that ratings have been well studied as they pertain to recommender systems. My work looks to examine user ratings in a new context, that of web-based tutorials.

Chapter 3

Measuring Quality

In this chapter, I discuss how I developed a list of measurable properties of tutorials that are indicative of a tutorial's quality; these properties are referred to as metrics. I begin by introducing a series of metrics that I found to be commonly used by websites to showcase the quality of tutorials; an example of such a metric is the average user-contributed rating of a tutorial. These showcasing metrics are summarized in Table 3.1. I next present metrics that I collected from the body of literature that exists on teaching computer skills through procedural instructions, deriving a number of metrics from what the authors determined to be important. Finally, I discuss how I augmented my list of metrics by interpreting comments posted by users about the tutorials. This second list of metrics is summarized in Table 3.2. I present these metrics in separate lists because I found, as will be discussed in section 4.4.1, that the showcasing metrics were not a good indication of a tutorial's quality.

3.1 Showcasing Metrics

This section focuses on the metrics that are commonly used by websites to help users choose between the available tutorials, which I list in Table 3.1. In assembling this list of metrics, I visited more than 50 different tutorial websites. I start with an overview of what types of metrics I found to be in use, and follow this up with a more in-depth discussion of each of these metrics. I also provide some examples of sites that use the different showcasing methods.

Some of the tutorial websites that I looked at only used the barest of information to showcase their tutorials. For example, www.photoshopessentials.com and www.adobetutorialz.com list their tutorials in an unspecified order with only a thumbnail of an image taken from the tutorial, its title, and a brief introductory paragraph to help users pick between them (see Figure 3.1 (a)). Others use combinations of ratings, numbers of social media shares, the number of times a tutorial was viewed, the date it was posted, and the name of the author or person who uploaded it (occasionally this is their real name, but most of the time it is their user name). Examples of sites that

Table 3.1: Showcasing Metrics

Metric	Description
Author	The user name or real name of the person who authored the tutorial.
Date Posted	The date that the tutorial was uploaded.
Rating	The rating that is displayed to the user; often a rounded average of all user-contributed ratings, with a maximum value of five.
Social Media Counts	The number of times a tutorial was advertised by readers using various social media platforms.
Views	The number of times a tutorial was viewed. This count sometimes includes repeat views by the same user.
Votes	The number of votes used in calculating the rating displayed to the users.

use several showcasing metrics include `psdlearning.com` and `psd.fanextra.com`, which use the date of posting, and the user-name of the author ((see Figure 3.1 (b)), and `www.good-tutorials.com`, which adds the number of views and a rating to that list of metrics (see Figure 3.1 (c)); this last website also includes the only example that I came across of a metric which indicates the number of users that have used their site to ‘save’ a tutorial that they would like to revisit in the future. Social media buttons, a slightly more common metric, but still fairly rare, can be found on the site `www.photoshoplady.com` (see Figure 3.1 (d)).

While ratings are not the most frequently available metric, they are the most direct attempt at conveying a sense of quality to the reader. These ratings are usually found on websites that collect tutorials from all over the web, such as `www.pixel2life.com`, but also appear on other sites, such as `psd.tutsplus.com`. In all cases that I found, the ratings appear as an image of up to five stars. The standard technique for submitting a rating is to select a number of stars from a row of five of them, allowing only full-star values. The rating that is displayed to the user is typically the mean of all submitted ratings rounded to the nearest half-star; in one case (`www.tutorialized.com`) I found that the rating was actually always rounded down from the mean. In short, the range of possible values for the rating is between one and five stars, with half-star ratings being possibilities.

I found that sites that included the average rating normally (but not always, such as in the case of `psd.tutsplus.com`) provide the number of votes that were collected in making up the rating. The purpose of this number is to lend legitimacy to the rating, increasing users’ confidence in the rating as the number of votes increases.

a) Surreal Motionscape Effect With Photoshop CS6

In this Photo Effects tutorial, learn how to add a surreal, otherworldly look to a photo with an easy-to-create motionscape effect!

Create a Futuristic Matte Painting Landscape

By [Vik Aadi](#), 27th May, 2013, [1 Comment](#)
(Photo Effects - Tutorials)

Today's design lesson will teach you how to create a futuristic matte painting landscape. You'll be working with a range of photo manipulation techniques including image extraction, image blending, adjustment layer usage, creating light effects, digital painting, advanced masking, color correction and many more. **Download the source file for this and all our other tutorials in our [FanExtra Members Area](#)**

[Read more...](#)

Best Way to Make Water Reflection in Photoshop!

[Photoshop](#) » [Photo Effects](#) — 2 months ago

Photoshop CS6 tutorial showing how to create a realistic water reflection of a skyline, landscape or object.

★★★★☆ 644 clicks 0 comments 1 save share report [bluelightningtv](#)

d) How to Create Realistic Human Hair

Under [Drawing Effect](#), [Photoshop Tutorial](#)

[Pin it](#) [Tweet](#) 7 [Like](#) 0

Painting hair is an essential step for making a portrait and there are various style of hairs. Therefore, this tutorial will teach you to process different hairs.

0 Likes [Add to Your Favorite](#)

Figure 3.1: Example showcasing metrics extracted from web pages: (a) www.photoshopessentials.com uses the bare minimum of showcasing metrics; (b) psd.fanextra.com includes the date the tutorial was uploaded, and the author; (c) www.good-tutorials.com adds number of views and an average rating; and (d) www.photoshoplady.com uses buttons for three popular social media sites.

In a similar vein to the ratings, I found a few tutorial sites allow users to click on a button to share a tutorial on social media platforms; an example of this is www.photoshoplady.com where buttons to ‘pin’, ‘tweet’, and ‘like’ the tutorials on Pinterest, Twitter, and Facebook, respectively (three popular social media sites), are provided. The Twitter and Facebook buttons also display the number of people who have already used the buttons to post the tutorial on their respective social media feeds.

A very common metric I came across in my survey of tutorial sites was the number of times that the tutorial has been viewed. This seems to rely on the idea that there is safety in numbers; the more a tutorial is looked at, the more likely it is to be good. Some sites, such as www.good-tutorials.com, excluded repeat visits from its count, while others, such as www.tutorialwiz.com, did not. One, www.psllover.com, counted three views every time a tutorial was viewed from their site.

The date the tutorial is uploaded was the most commonly available metric; this provides users with additional context for the popularity of the tutorial (by cross referencing it with the number of votes and views), and gives the user a general impression of how current to expect the version of Photoshop used in the tutorial to be.

A metric that I came across with almost the same frequency as the date was the user name of the author of the tutorial. The real name was occasionally included instead, such as in the case of photoshoptutorials.ws. This seems to be available so that readers who like a tutorial can search for tutorials by the same author (or ignore tutorials by disliked authors); it serves to establish a reputation for the author.

The metrics listed above are ones that are used in practice to try to help users make the decision as to whether they should spend time reading a tutorial. Next, I discuss metrics that I collected from the literature on teaching computer skills through procedural instructions.

3.2 Metrics from Related Work

In this section, I discuss how I used the literature on teaching computer skills to users to derive 21 of the metrics that I will use in section 4.4 for discussing the quality of tutorials. When referring to metrics throughout this section, I will include the ID's that are found in Table 3.2 in brackets for easy cross-referencing. The metrics that I discuss in this section are $M1$ – $M21$; the others (derived from users' posted comments) will be discussed in section 3.3.

Booher [4] and Stern [58] found that a combination of printed words ($M1$) and images ($M2$) were required for tutorials to be effective; images allowed for speed, while text was necessary for accuracy. While strictly speaking this implies that the presence of words and images are important, the metrics that I use are the numbers of these things; this allows me not only to discuss whether they are present, but also how they relate to other metrics. For instance, in chapter 4 I will discuss the number of images in relation to the number of steps to get a sense of how often written instructions are accompanied by visuals.

Grabler et al. [21] had a series of suggestions for creating high-quality tutorials, attributing the design principles to studies conducted by several authors ([2, 4, 26, 29, 48]). Based on their survey of prior studies, Grabler et al. indicated that tutorials

Table 3.2: Metrics. ID's are provided for easy cross referencing. Bibliographic sources are listed after the metric name; if a source is not listed for a metric then it was derived from user comments (see section 3.3). Rules for defining subjective metrics are also provided.

ID	Metrics and Sources	Additional Collection Information
<i>M1</i>	Number of images [4, 58]	Automated collection; manually removed ad images
<i>M2</i>	Number of words [4, 58]	Automated collection; manually removed unrelated headers and ad text
<i>M3</i>	Is step-based [21]	Delimits segments using white space, images, numbers, or bullets
<i>M4</i>	Has numbered steps [21]	Objective
<i>M5</i>	Number of steps [21]	Objective (given <i>M3</i>)
<i>M6</i>	Has end of step images [21]	Has screenshots of relevant interfaces or expected workspace at the end of some (other than the last) steps
<i>M7</i>	Number of end of step images [21]	Objective (given <i>M6</i>)
<i>M8</i>	Number of textual references to images [21]	Objective
<i>M9</i>	Number of images with tool palettes [21]	Any image of a toolbar or palette with a tool selected; unselected toolbars were not counted
<i>M10</i>	Number of images with parameters [21]	Images with any values filled into text boxes, check boxes, etc.
<i>M11</i>	Number of parameters in text [21]	In-text values for text boxes, check boxes, etc.
<i>M12</i>	Number of images with annotations [21]	Objective
<i>M13</i>	Number of annotations in images [21]	Objective
<i>M14</i>	Repetitive steps condensed [21]	References past steps from similar steps
<i>M15</i>	Number of tips and hints [7]	Text that describes common problems and what can be done to mitigate those problems
<i>M16</i>	Number of explanations of why steps are conducted [7]	Text that describes why a step was taken (not just how to do it)
<i>M17</i>	Number of textual references to shortcuts [30]	Objective
<i>M18</i>	Number of textual references to menus [30]	Objective
<i>M19</i>	Number of textual references to tools [30]	Objective
<i>M20</i>	Presence of final image [30]	Objective
<i>M21</i>	Preview of final image [30]	Final image appears at the start of the tutorial, not just at the end
<i>M22</i>	Presence of source files	Includes a list of source files (just a starting image, if no other resources are used) and where to download them
<i>M23</i>	Presence of original image	Objective
<i>M24</i>	Specified version	Objective

should use numbered steps (*M3–M5*), and combine text descriptions of the actions to take with screenshots of the results of completing the step (*M6–M8*). They also specified that screenshots should contain relevant interface widgets (*M9*), with any needed parameters either filled in (*M10*), or specified in the description of the steps (*M11*). The authors also advised that annotating images increases understanding (*M12–M13*), specifically suggesting the use of arrows and highlighting to denote areas of interest. Finally, they said that repetitive steps should be condensed through references to past steps (*M14*).

Carroll [7] ran a series of studies on users' practices with learning new software. From these studies, he suggests that helping users with potential errors should be included in any instructional material (*M15*). He also describes how users have a tendency to not read material fully, and to follow tutorials without understanding why they are doing things. The results of this are an inability for users to consistently use learned techniques in different situations than they were originally learned. Consequently, he places high value on explanations describing why steps are important (*M16*).

In their formative study of the Delta tool, Kong et al. [30] found that when searching for a tutorial, users emphasized the importance of knowing which commands are used in the tutorial (*M17–M19*), and being able to ascertain the results of following the tutorial (*M20–M21*).

These metrics have all been shown in the literature to be important in instructing users to learn computer systems. In the next section, I will discuss the final source that I used for collecting metrics, comments posted by readers of the tutorials.

3.3 Metrics from Users

The comments that most sites allow readers to post at the end of a tutorial are a great source for discovering what users find particularly useful about tutorials. The comments that I based the balance of the metrics on were collected from a series of tutorials that were collected as part of published work that I conducted with colleagues [33]. A subset of these tutorials makes up the set of popular task-based tutorials that I describe in section 4.1.4.

From these comments, I found that users often would ask for the location of the source files that are used in the tutorial ($M22$ – $M23$), as well as which version of Photoshop was being used ($M24$), as each version results in significant changes being made to the software’s functionality and interface.

Other metrics that have already been discussed in section 3.2 were also frequently brought up in user comments. This includes requests for authors to annotate areas where tools were used ($M12$, $M13$), and questions surrounding what parameters were entered in dialogs ($M10$, $M11$). One comment from a user expressed a dislike for single-paragraph and single-image tutorials ($M1$, $M3$, $M5$).

3.4 Summary

In this chapter I presented a series of metrics for measuring the quality of tutorials. These metrics came from a mix of the literature on procedural instructions, comments posted at the end of tutorials, and the techniques currently used by websites to showcase the quality of tutorials.

Chapter 4

A Study of Online Photoshop Tutorial Quality

In this chapter I describe the study that I conducted to inform my discussion of the quality of online tutorials. This study focused on selecting a series of 154 Photoshop tutorials from four online sources, and then collecting values for each of the 24 metrics listed in Table 3.2 for each of these selected tutorials. I also collected values for the showcasing metrics listed in Table 3.1 for the 18,133 tutorials present on one of the source websites. My analysis of this data frames my discussion of the adherence of the tutorials available online to principles of effective tutorials (as described in prior work, as well as by users via their comments), addressed both, as a single group, and between four different tutorial sources. I also show that methods that are currently used by websites to showcase quality of tutorials are not effective.

This chapter begins with a description of the different sources that I collected tutorials from. I then move onto how I selected the tutorials from those sources, and

discuss my process for collecting values for the metrics. I then present the results of my analysis of this data in three parts: the utility of current showcasing methods; the overall quality of the tutorials in my sample; and the difference in quality of tutorials from different sources. I conclude this chapter with discussions of the results and limitations of the study.

4.1 Tutorial Sources

In conducting my initial survey of Photoshop tutorial sites (previously discussed in section 3.1), I found that most of the sites had characteristics that allowed me to group them into three categories: application-centered communities, tutorial aggregators, and tutorial factories. Because the differences between these groups result in different audiences and expertise of authors, I chose one source from each of these groups to sample tutorials from. In addition to these sources, I also chose a set of popular task-based tutorials so I could characterize the tutorials that users are likely to be viewing when seeking help with a specific task; these tutorials were collected through a technique known as CUTS [18]. Below, I describe the properties of these four sources of tutorials.

4.1.1 Application-Centered Community

An application-centered tutorial community is a tight-knit group of application-specific enthusiasts who share their knowledge and skills with other members of the community, and often the public at large. Members of these communities range in experience from new users, to those who use the application as part of their profes-

sions, with tutorials being written in their spare time. The tutorials are not offered for a price, although authors may be profiting indirectly through related services. These websites are generally more than simply lists of tutorials, also providing forums, member services, and private messaging facilities.

I chose Renderosity (www.renderosity.com) as a site representative of this type of community. Renderosity is a site that helps digital artists (artists who use electronic media) connect with one-another, allowing them to discuss and share techniques. Renderosity hosts galleries for showcasing members' art, member profiles, forums, member-submitted tutorials, member-submitted electronic-art supplies (such as custom textures and brushes for Photoshop), and a marketplace for selling members' art. It uses the number of views, a link to the author's member page, and the date that the tutorial was posted to showcase the tutorials. At the time of my study, Renderosity had a total of 126 Photoshop tutorials available.

4.1.2 Tutorial Aggregator

Tutorial aggregator sites typically do not host the actual tutorials, but link to as many as they can, and generally for a variety of applications. This means that members of these communities do not necessarily write the tutorials that they post, but that the sites have a wide sampling of the tutorials that are available across the web. It is unclear if community members are paid to collect tutorials, or if they collect them for more altruistic reasons. Interactions between member users are limited; the term community is used very loosely for this type of site.

I chose Tutorialized (www.tutorialized.com) as my tutorial aggregator. Of

the showcasing strategies I've discussed previously (listed in Table 3.1), Tutorialized doesn't employ the use of social media counts; this isn't very surprising since I encountered these counts in only a few of the websites. It also doesn't list who provided the links to the tutorials; I expect that the individuals operating the site seek out the tutorials. Tutorialized linked to 18,133 Photoshop tutorials at the time of my study, more than any other site that I examined.

4.1.3 Tutorial Factory

A tutorial factory is a site run by a company that pays people to author tutorials. My observations of these sites indicate that the authors are typically expert users of Photoshop, either involved in developing the application itself, or in using it in their daily lives. These sites only post tutorials that they accept, but they usually do not advertise what their acceptance criteria is. In examining the tutorials on these sites, I found they frequently end up with a small set of authors who write tutorials for them.

I chose PhotoshopTutorials (www.photoshoptutorials.ws) for this type of tutorial community. The site pays authors between \$150 and \$300 per tutorial, and includes short biographies discussing the authors' credentials and expertise. At the time of the study they used authors' real names (with links to their online profiles), and posting dates, to showcase the tutorials. They had 493 tutorials available.

4.1.4 Popular Task-Based Tutorials

For the fourth source, I used the CUTS technique [18] to generate the three most common “how to” searches pertaining to Photoshop. Searching for tutorials related to these queries provides a source of popular task-focused tutorials. This source is meant to represent the tutorials that an average user is most likely to view when searching for help with completing a task with Photoshop by using Google (www.google.ca). I elaborate on my use of CUTS when describing my data collection method in section 4.2.2.

4.2 Tutorial Sampling

This section focuses on how I selected tutorials from the four sources discussed above; my sampling procedure differed between the tutorials used in analysis of the two sets of metrics.

4.2.1 Selecting Tutorials for Showcasing Metric Analysis

For collecting values for the showcasing metrics, I used tutorials only from Tutorialized. The tutorials from neither Renderosity, nor PhotoshopTutorials, contained ratings, and they were found in only some of the tutorials selected using the CUTS method. Additionally, Tutorialized has succeeded at collecting such a large number of tutorials that I felt that it should contain a representative range of the tutorials that are available online.

Because the showcasing metrics were easy to collect (as I describe in section 4.3.1),

I chose to use all 18,133 of the available tutorials from Tutorialized.

4.2.2 Selecting Tutorials for Quality Metric Analysis

Because these metrics are quite time consuming to collect (see section 4.3.2), I could not collect values for my metrics from every available tutorial. For three of my sources, I randomly sampled a subset of the available tutorials: 20 from the 493 available on PhotoshopTutorials, 20 from the 126 available on Renderosity, and 100 from the 18,133 available on Tutorialized. Below, I describe how I selected 14 tutorials from the fourth source.

CUTS

I used the Characterizing Usability Through Search (CUTS) method [18] as described by Lafreniere et al. [33] to supplement the sample of tutorials with 14 tutorials that address some of the most common issues searched for by users. I go into details about how it works below.

The CUTS method combines Google’s query completion suggestion service (Google Instant) with their web-based advertising tool (Google AdWords) to determine an ordering of search queries reflective of the frequency with which the queries were made to Google.

In addition to the description of the CUTS method, Fourney et al. [18] also provide a classification of query phrasing, linking the style of phrasing to the intent of the search. Of particular note was the ‘question’-style query which users used to find operating instructions and reference materials (i.e., tutorials). This style of query

had the form “how to ...in System Name”.

The CUTS method relies on the fact that Google Instant provides suggestions of the most frequent queries using all the text already typed into the search bar, and Google AdWords provides frequency counts for a subset of the possible queries; the suggestion database is updated frequently with current search queries being added, and older queries and those with declining popularity removed.

Next, an (automated) exhaustive Google search of all queries matching the form of the ‘question’-style query that returned any suggestions from Google Instant was performed; that is, queries starting with “how to ”, and ending with “ in Photoshop”. The substring between the two phrases was made up of any sequence of characters that returned suggestions from Google Instant. Google AdWords is then used to order the suggested queries by the frequency that they were searched for.

The three highest ranked queries were:

1. how to cut out an image in Photoshop
2. how to feather in Photoshop
3. how to make a mix tape cover in Photoshop

For each of the above searches, I selected the first five unique tutorials that Google returned. I sampled a total of 14 tutorials using this approach; the third query returned only four unique tutorials within the first five pages of results (with ten results on each page), and it seemed unlikely that a user would search further than that. This brought my total sample size up to 154.

4.3 Data Collection Method

This section focuses on the methods that I employed in collecting the values for the metrics described in chapter 3. First, I describe the collection of values for the showcasing metrics, and then move onto the others.

4.3.1 Showcasing Value Collection

When choosing a tutorial to view on Tutorialized, it lists ten tutorials at a time, with showcasing values for each one visible without having to request the actual tutorial. To collect values for the showcasing metrics from the 18,133 tutorials available on Tutorialized, I wrote a script that requested each of the 1,814 pages needed to view all of the tutorials' showcasing values, and parsed the HTML for the relevant data. I collected the tutorials' ratings, the number of votes involved in those ratings, the number of times the tutorial was viewed, and the date the tutorial was posted.

While the ratings that are displayed to the users on Tutorialized are displayed as images that are discrete to half a star, the HTML code actually makes the rating available to two decimal places. From this, I discovered that Tutorialized rounds their ratings down to the next half-star before displaying them to the users. I collected these more accurate ratings, but because users of the website most likely do not view the raw page source, I analyze only the ratings as they are displayed to the user: rounded down.

4.3.2 Metric Value Collection

I manually examined the content of each tutorial in my sample and collected values for each of the 24 metrics listed in Table 3.2. This table also includes a short-form list of the rules I followed when collecting values for the metrics; a more detailed explanation is available below. Each tutorial took approximately one hour to fully analyze.

Values for the number of images ($M1$) and number of words ($M2$) were collected via an automated system. For the images, I wrote a script that searched through the HTML for image tags. For the number of words, I pasted the text from the tutorials into Microsoft Word, and used its automatic word counter. I removed the effects of advertisements and other unrelated information manually; this involved going through each of the tutorials, and reducing the number of words and images by the number of each that were found in unrelated content. All of the values for the other metrics were collected manually.

Of the 24 metrics, 7 of them were objective, requiring no special rules for collecting values for them: whether the tutorial used numbered steps ($M4$); number of textual references to images ($M8$), shortcuts ($M17$), menus ($M18$), and tools ($M19$); number of images with annotations ($M12$); and whether they specified a version of the tutorial ($M24$).

The other metrics required rules for value collection. I determined if a tutorial was step-based ($M3$) by checking whether it used any means to separate the text into sections. In practice, authors used white space, images, numbers, and bullets to separate the text into steps; some authors, however, used a single large block of text,

either running along beside images, or on its own. I use the same delimiting rule for determining the number of steps (*M5*).

I consider an end-of-step image to be any image that has screenshots of the workspace at the end of a step. This may be an image of an edited photo, or of an interface with parameters filled in; it depends on what the text of the step actually entailed. I used this definition to find whether tutorials had them (*M6*), and how many they had (*M7*).

To determine the number of images with tool palettes (*M9*), I had to make sure that they were present in the images in order to provide additional information for the users, and not simply included as a result of a screenshot. For example, the main tool palette was often included in screenshots of the main work area even though it was not used in the related step. Thus, I only counted any image that included a toolbar or tool palette with a tool selected or otherwise emphasized; this emphasizing often came in the form of annotated highlighting or circling of tools.

For an image to be considered to have a parameter in it (*M10*) the image had to include any interface component that has a value assigned to it; examples include text entered in text fields, checked check boxes, and selected items in list boxes. When counting the number of parameters in the text (*M11*), I similarly included mentions of any value for any interface component.

When counting the number of annotations in the images of the tutorials (*M13*), I considered a separate annotation to be any instance of annotation that is providing different information from the other annotations in an image; it was rare that I needed to make this determination, but occasionally the tutorials would contain things such

as two circles around the same point of interest. I originally collected separate numbers for each of the types of annotations that I came across (text, arrows, circles, and highlighting), but there appeared to be no statistical difference between the different types of annotations, and the data for each type was very sparse; consequently, I amalgamated them, and only include data and analysis for the total number of annotations.

I determined whether a tutorial had instances of condensing its repetitive steps (*M14*) by checking to see if at any point in the tutorial it instructed users to do something in the same way as was done in an earlier step by directly referencing that earlier step. If I was unable to find any instances of repeated instruction, I recorded it as such; as a result, there were three possible values for this metric (yes, no, and none).

I considered any text that described possible problems and potential ways to get around them to be a tip (*M15*); this had to be in addition to what was necessary to describe how to complete the step in the case that there were no complications. For counting the number of explanations of why a step was taken (*M16*), I looked for an explicit reason for taking a step, rather than simply an explanation of how to complete it.

I judged a tutorial to contain source files (*M22*) in the event that it contained links to every file that was necessary to complete the tutorial exactly as described by the tutorial. Sometimes this required an explicit link, such as in the case of custom-made brushes, but in the case of images, simply including the image in the file such that the user can save the image (without having to modify it) in order to use it was sufficient.

Including a screenshot of an image within the application's interface was not sufficient because the reader would need to extract the desired image before they could use it. This could impact the quality of the image, and could make progressing with the tutorial difficult. This same rule was used in determining whether the author had met three metrics concerning inclusion of special source files: an image of the results of the tutorial (*M20*), whether that same image is shown as a preview at the start of the tutorial (*M21*), and any image that is required to begin the tutorial (*M24*).

4.4 Results

In this section I present my analysis of the quality of my sample of online tutorials. I first show that the current showcasing metrics that are used are not giving an accurate picture of the quality of a tutorial. Next I discuss where the tutorial authoring community as a whole is succeeding and struggling, and then move onto a dissection of the differences in quality of tutorials from my four tutorial sources, showing that the expert-authored tutorials of PhotoshopTutorials are succeeding in a number of ways that the others are not. I end this section by discussing these results, and presenting the limitations of my study.

4.4.1 Current Showcasing Strategies

The presence of showcasing methods is an attractive way of determining the quality of a tutorial without having to examine it; my analysis of the data on the showcasing metrics that I collected from Tutorialized, however, indicates that these metrics are not a good way of judging quality.

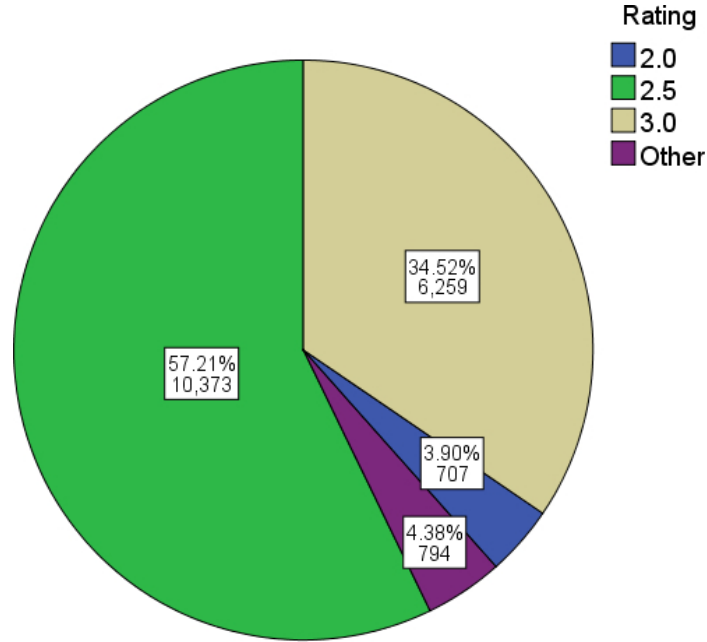


Figure 4.1: Ratings for all tutorials from Tutorialized ($n = 18,133$). Ratings attributable to less than 2% of the total number of tutorials have been collapsed.

Take for example the average rating displayed to users. Tutorialized rounds the rating down to the next half star on a five-star scale. Because it is impossible to rate a tutorial at lower than 1.0 stars, there are nine possible values for the displayed ratings. Figure 4.1 shows that out of the 18,133 ratings I collected, 16,632 (91.7%) were either 2.5 (57.2%) or 3.0 (34.5%); another 707 (3.9%) had the rating of 2.0, resulting in 95.6% of the ratings being clustered into three of the nine possible values. This implies that there is nearly no diversity of quality, which, besides being highly unlikely, my personal observation can discredit. Therefore, the ratings are not functioning as accurate discriminators.

The number of votes is supposed to add confidence to a tutorial's ratings, but, as can be seen in the boxplots of Figure 4.2, the vast majority of the tutorials with ratings other than 2.5 and 3.0 have very few votes. For the sake of legibility, the

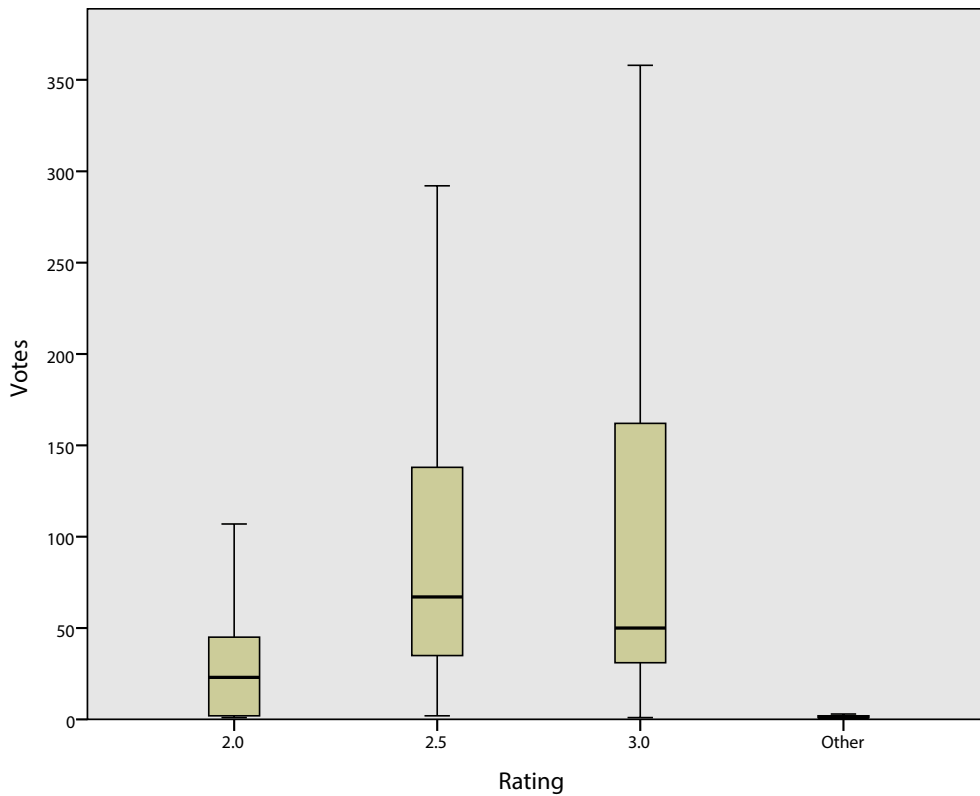


Figure 4.2: Vote boxplots separated by rating. Ratings attributable to tutorials with less than 1% of the total votes have been collapsed. For legibility, outliers and probable outliers, as defined by Tukey [60], have been removed.

ranges depicted in this graph have had outliers and suspected outliers removed using Tukey’s criteria [60]. This criteria defines suspected outliers as tutorials with votes higher than 150% of the IQR above the third quartile, or lower than 150% of the IQR below the first quartile. I have included the medians and IQR’s used for this chart in Table 4.1. I found that the number of votes is significantly correlated with the date that the tutorial was uploaded ($p < 0.001$, $R^2 = 0.493$). From this, I infer that the number of votes is at least partially representative of how old the tutorial is, and increasing votes has a centralizing effect on the rating.

Table 4.1: Number of votes by tutorial rating ($n = 18,133$).

Rating	Percent of Tutorials	Median Number of Votes	Inter-Quartile Range of Votes
1	1.1	1	0
1.5	0.2	1	10
2	3.9	23	43
2.5	57.2	67	103
3	34.5	50	131
3.5	0.7	22	29
4	1.2	1	0
4.5	0.1	2	1
5	1.1	1	0

The number of views appeared to be at least equally as problematic as the votes, since the number of views is confounded by the amount of time the tutorials have been online, activity on the site during that time, and by what are evidently misleading ratings.

The above results show that the showcasing methods that are currently available to users to distinguish between quality of tutorials are not providing accurate information: they're implying that nearly all of the tutorials are of equivalent, medium-level, quality, with primarily new tutorials being seen as either high- or low-quality. By providing the number of times that a tutorial has been viewed as a judge of quality, this inaccuracy is intensified.

4.4.2 Tutorial Quality

In presenting my examination of the quality of the tutorials, I begin by looking at the overall quality, and then turn to a discussion of the differences in quality

between the tutorial sources. To compensate for the variability in the length of the tutorials and complexity of the tasks associated with them, I normalized the quantitative metrics by dividing by the number of steps. There were three exceptions (*M17–M19*); since experience with commands, menus, and shortcuts are integral to promoting expertise of the system [43], I analyzed their raw counts. A subset of the analysis of the overall quality of the tutorials was included in a recent publication [33].

Quality Across All Tutorials

Below, I describe the results by dividing my discussion into what tutorials are doing well, where the results are mixed, and where the tutorials are falling short; these results are summarized in Table 4.2. Overall medians and inter-quartile ranges (IQRs) for the quantitative metrics can be found in Table 4.3 (column six: 'All'), with percentage values for categorical metrics listed in Table 4.4.

Where Tutorials Are Adhering to Guidelines When examining values for the quality metrics in relation to previously established guidelines, I see that there are numerous things that the majority of tutorials appear to be doing right.

Table 4.2: Summary of overall quality of tutorials.

Quality	Metrics
High	Number of images (<i>M1</i>), images with parameters (<i>M10</i>), references to parameters (<i>M11</i>), repetitive steps condensed (<i>M14</i>), final image (<i>M20</i>), source files included (<i>M22</i>), original image (<i>M23</i>)
Mixed	Tips and hints (<i>M15</i>), references to shortcuts (<i>M17</i>), references to menus (<i>M18</i>), references to tools (<i>M19</i>)
Room for Improvement	Numbered steps (<i>M4</i>), textual references to images (<i>M8</i>), annotations (<i>M12</i> , <i>M13</i>), Explanations (<i>M16</i>), previews of final image (<i>M21</i>), version present (<i>M24</i>)

For example, 141(91.6%) of the tutorials contained the original images (*M23*) that were being worked on, and 148(96.1%) showed the final image (*M20*) that the tutorial produced. With both of these included in a tutorial, the user is more easily able to tell what the effect of following the tutorial will be, and to verify that they completed the tutorial correctly.

Of the 154 tutorials collected, 130(84.4%) had at least one image per step (*M1*). This means that most of the time, users are looking at tutorials that have images that go with the instructions, helping them to follow along. In the same vein, the number of textual references to parameters per step (*M11*) (median=0.5, IQR=0.89) is generally higher than the number of images that contain parameters per step (*M10*) (median=0.4, IQR=0.59). Because of the significant range in values, I decided to look at this on a case-by-case basis; I found that in 102(66.2%) of the tutorials, there were at least as many references to parameters as there were images with parameters. This implies that the authors are likely referencing parameters in the instructions that appear in the images as well. This type of redundancy is considered to be positive, since the two styles together allow for both speed, and accuracy for those trying to follow the tutorials [4, 58].

Tutorials also tended to refer back to previous steps when possible (*M14*) (60/67 or 89.6% of tutorials containing repeated steps), which can result in better memory retention for the reader [7].

Finally, 90.3% of the tutorials included links to all the source files needed to complete the tutorial (*M22*). Without these links, users need to come up with their own resources; unless they can mimic these resources exactly, they are unlikely to be

able to verify that they're following the instructions correctly as they complete the tutorial.

Mixed Quality My analysis also reveals some strong variation; while many tutorials are following the established guidelines, there is also a great deal of room for improvement.

An example of this is the number of tips and hints per step ($M15$) (median=0.11, IQR=0.25). The presence of tips and hints is an indication that the authors are considering the problems that may be encountered by the audience. With at most 0.11 tips and hints per step for half the tutorials, and 61(40.7%) tutorials not containing any at all, authors seem to be taking for granted their ability to convey instructions to the users. Examining the third quartile (0.25), however, shows that a quarter of the tutorials do provide this type of assistance for at least a quarter of their steps.

Similarly, the textual metrics concerning direct interaction with the user interface ($M17$ – $M19$), exposure to which is known to be an integral component to expertise [43], all have relatively low median values (1, 2.5, and 4.5 respectively). This indicates that these tutorials may be limited in their ability to increase a user's exposure to the application's command set (the more of the command set that is used, the more likely it is that the reader is exposed to something they've never used before). That being said, much like the number of tips and hints per step, the third quartile (5, 6, and 12 respectively) reveals that many of the tutorials potentially provide quite a bit of exposure to commands.

Table 4.3: Analysis of quantitative metrics. All metrics are normalized by step, except *M5* and *M17–M19*. Columns 2–5 represent the sources: P = PhotoshopTutorials, R = Renderosity, G = Google, T = Tutorialized. Values for a source that are significantly higher than all others are in bold.

Metrics	Median (IQR)					Main Effect		Pairwise significance ($p < 0.05$)
	P	R	G	T	All	H	$p <$	
Images (<i>M1</i>)	2.07 (2.39)	1.00 (0.63)	1.07 (0.30)	1.50 (0.79)	1.35 (0.96)	23.304	0.001	$P > R, P > G,$ $T > R$
Words (<i>M2</i>)	68.40 (85.47)	46.91 (126.42)	44.65 (36.28)	37.08 (38.17)	43.37 (45.92)	12.716	0.01	$P > T$
Steps (<i>M5</i>)	14.50 (21.50)	12.00 (15.25)	7.00 (8.00)	7.00 (8.00)	1.35 (0.96)	12.512	0.01	$P > T$
End of Step Images (<i>M7</i>)	0.87 (0.28)	0.69 (0.49)	1.00 (0.22)	0.57 (0.53)	0.67 (0.51)	18.991	0.001	$P > T, G > T$
References to Images (<i>M8</i>)	0.65 (1.09)	0.34 (0.81)	0.17 (0.55)	0.31 (0.54)	0.33 (0.67)	4.992	NS	NS
Images with tool palettes (<i>M9</i>)	0.88 (1.48)	0.27 (0.73)	0.33 (0.38)	0.56 (0.62)	0.56 (0.68)	17.479	0.001	$P > R, P > G$
Images with parameters (<i>M10</i>)	0.74 (1.34)	0.21 (0.59)	0.09 (0.25)	0.40 (0.57)	0.40 (0.59)	27.450	0.001	$P > \text{All},$ $T > G$
References to parameters in text (<i>M11</i>)	1.31 (3.45)	0.40 (0.84)	0.17 (0.45)	0.46 (0.77)	0.50 (0.89)	25.019	0.001	$P > \text{All}$
Images with annotations (<i>M12</i>)	0.09 (2.25)	0.00 (0.17)	0.00 (0.16)	0.00 (0.28)	0.00 (0.24)	2.714	NS	NS
Annotations (<i>M13</i>)	0.11 (3.64)	0.00 (0.28)	0.00 (0.21)	0.00 (0.42)	0.00 (0.38)	3.072	NS	NS
Tips and hints (<i>M15</i>)	0.23 (0.24)	0.16 (0.80)	0.27 (0.45)	0.00 (0.17)	0.11 (0.25)	29.316	0.001	$P > T, R > T,$ $G > T$
Explanations (<i>M16</i>)	0.22 (0.49)	0.17 (0.54)	0.07 (0.33)	0.00 (0.17)	0.05 (0.29)	24.656	0.001	$P > T, R > T$
References to shortcuts (<i>M17</i>)	7.50 (16.50)	0.00 (3.00)	0.00 (4.25)	1.00 (4.00)	1.00 (5.00)	15.784	0.001	$P > \text{All}$
References to menus (<i>M18</i>)	7.50 (9.00)	2.00 (4.75)	1.00 (3.00)	2.00 (4.00)	2.50 (5.00)	21.193	0.001	$P > \text{All}$
References to tools (<i>M19</i>)	18.00 (16.75)	4.50 (11.75)	4.00 (10.75)	3.50 (7.00)	4.50 (10.00)	25.350	0.001	$P > \text{All}$

Room for Improvement My analysis reveals that web tutorial authors are falling short in a few areas. The most obvious example concerns the number of explanations per step (*M16*) (median=0.05, IQR=0.29). This is likely to impact long-term learning, as it can help users both, call upon past experience to more readily under-

stand what needs to be done, and apply the current steps to similar situations they encounter in the future [7]. While explaining every step is likely excessive, approximately only one in every twenty steps was explained.

I also found that only 72 (47.8%) of the tutorials contained annotations in any of their images (*M13*). Annotations can help describe things that are much harder to convey through textual instruction alone, such as where to apply brush strokes, or what areas of the image to select. They can also be used to highlight small changes that may otherwise be missed by someone completing the tutorial.

Most authors are also not including what version of Photoshop the instructions work for (*M24*), with only 26 (16.9%) of the tutorials containing version information. Version-related questions were fairly common in the comments, either asking how to complete a step in a given version, or asking what version was being used. This can impact whether a reader can complete the tutorial at all, as many of the more specialized commands have been introduced in recent versions of Photoshop; if it is possible to find a workaround for one of these commands, it may have a very complex algorithm.

One of the more surprising things that I saw in my analysis was that authors frequently did not number their steps (*M4*), with only 93 (60.4%) of the tutorials doing so. This makes it difficult to refer back to specific steps, either by the author within the tutorial, or by the users afterwards, in the comments. Numbering the steps can have other impacts as well, such as giving users a general idea of how long a tutorial will take, or how much of it is left.

While users are unlikely to invest the time into a tutorial if they cannot see the

effect that they will be creating ahead of time, authors often failed to include previews of the final image (M_{21}), with only 85 (55.2%) of the tutorials having them. This can be a frustration to the reader because they may need to navigate to the end of the tutorial before they can see what the final image will look like; this can require several page navigations, as many sites break long tutorials into multiple pages.

Quality Differences Between Communities

While in the previous section I examined tutorial quality across all four sources, in this section I examine quality differences between the sources. Since the sampled data produced right-skewed histograms for all of my quantitative metrics, I performed the non-parametric Kruskal-Wallis test for effects of tutorial source on them. Post-hoc pair-wise comparisons were corrected using Bonferroni corrections. Table 4.3 provides the relevant statistics, presenting medians and IQRs for each group, and summarizing the significant pair-wise comparisons.

As Table 4.3 illustrates, only three of the main effects were not significant: the number of textual references to images per step (M_8), annotations per step (M_{13}), and images with annotations per step (M_{12}).

For the metrics with significant main effects, pair-wise comparisons showed that tutorials from PhotoshopTutorials were better than or equivalent to the remaining tutorial sources: the tutorials from PhotoshopTutorials were superior to those from Tutorialized in 10 of the 15 quantitative metrics (M_2 , M_5 , M_7 , M_{10} , M_{11} , and M_{15} – M_{19}), and to those from both Renderosity and Google in seven (M_1 , M_9 – M_{11} , and M_{17} – M_{19}).

Table 4.4: Analysis of categorical metrics: Pearson’s chi-square counts. Observed counts that differed significantly from expected counts are displayed in bold. Columns 6–9 represent the sources: P = PhotoshopTutorials, R = Renderosity, G = Google, T = Tutorialized.

Metrics	% “Yes”	X^2	$p <$	Counts	P	R	G	T
Step Based ($M3$)	97.4	2.9	NS	Yes Expected	20 19.5	20 19.5	13 13.6	97 97.4
Numbered Steps ($M4$)	60.4	24.2	0.001	Yes Expected	20 12.1	5 12.1	7 8.5	61 60.4
Repetitive Steps Condensed ($M14$)	89.6	17.6	0.01	Yes Expected	16 7.8	8 7.8	4 5.5	32 39.0
Final Image Present ($M20$)	96.1	9.1	0.05	Yes Expected	19 19.2	19 19.2	11 13.5	99 96.1
Preview of Final Image ($M21$)	55.2	14.7	0.01	Yes Expected	19 11	10 11	7 7.7	49 55.2
Source Files Present ($M22$)	90.3	8.9	0.05	Yes Expected	20 18.1	17 18.1	10 12.6	92 90.3
Original Image Present ($M23$)	91.6	2.9	NS	Yes Expected	19 18.3	18 18.3	11 12.8	93 91.6
Specified Version ($M24$)	16.9	14.6	0.01	Yes Expected	8 3.4	3 3.4	5 2.4	10 16.9

While PhotoshopTutorials was significantly better than the other sources for a number of metrics, the data did not suggest a clear ordering among the remaining three sources. This was especially true for Google and Renderosity, where the pairwise comparisons revealed no significant differences between the two sources. Google and Renderosity were significantly better than Tutorialized in two metrics ($M7$ and $M15$, and $M15$ and $M16$ respectively), while Tutorialized was higher than Renderosity and Google in just one metric each ($M1$ and $M10$ respectively).

For the categorical data, I calculated Pearson’s Chi Square values and likelihood ratios, and found significant differences between the sources in six of them ($M4$, $M14$, $M20$ – $M22$, and $M24$). Table 4.4 summarizes these results, and lists both, the actual and expected counts. I present only the “Yes” counts for the sake of simplicity.

In examining the differences between the actual and expected counts, the most striking differences are often found with PhotoshopTutorials. In all of its significant cases ($M4$, $M14$, $M21$, and $M24$), the observed counts are close to double the expected. Google and Tutorialized, on the other hand, had lower than expected counts for a number of metrics ($M20$ and $M22$; and $M14$, $M21$, and $M24$ respectively), with each having higher than expected counts for one metric ($M24$, and $M20$ respectively). Renderosity had one significantly lower than expected count ($M14$), and no higher than expected ones.

4.5 Discussion

I begin my discussion by examining the results concerning current showcasing strategies, presenting an argument for multidimensional rating systems. I then present an examination of the areas in which tutorial authors can use assistance, and conclude with an exploration of the apparent benefits for readers that can be found by selecting tutorials from expert authors.

4.5.1 Showcasing Strategies

One of my more interesting findings is that the showcasing methods on Tutorialized do not appear to be serving their intended purpose. The ratings do not vary enough to provide users with any power to perform between-tutorial comparisons.

This finding means that users currently do not have an effective way to narrow down the list of tutorials without examining their actual content. The delta tool [30] partially addresses this issue by depicting a tutorial's work-flow and command set.

This information, however, might not be sufficient to help the user judge the tutorial's instructional quality.

One possible explanation for this lack of accuracy is that it is difficult for users to determine what to base their ratings on, since what makes for a good tutorial is multidimensional, and user-dependent. Comments at the end of tutorials suggest that some users might be rating the tutorial based on how easy it is to follow, while other users seem to be considering the final effect, and how long it took them to complete it. This inequality in what is being measured would have an unpredictable effect on the resulting measurements. User expertise is also likely a factor; what is considered to be easy to follow by an expert user might be quite confusing to a novice. Future research should be conducted concerning whether multidimensional rating systems can produce a more accurate measure of quality.

4.5.2 Quality Across All Tutorials

My data indicates that while many tutorials are following principles of effective tutorial design, there were also some notable omissions by the tutorial authors, revealing opportunities for tutorial authoring systems. Among these omissions is a lack of inclusion of both, explanations for why a step should be undertaken, and tips for completing more difficult steps. Interestingly, despite the recent surge in automatic tutorial authoring tools (e.g., [8, 10, 21]), I have not come across any that have focused on these aspects, except to say that authors could add this information after the tutorials are generated (e.g., [8]).

Annotations are another missed opportunity for tutorial authors. Annotations

allow authors to easily show screen-location information, such as the path to move your mouse when using the sharpen tool, or the exact spot a blur should be focused on. In many cases, annotations allow still images to convey as much information as a video clip would, without the overhead for the reader that is associated with scrubbing through a video's timeline.

Other problems that I found would have low overhead for authors (a low incidence of numbering steps, inclusion of version information, and previews of the final image) and are all things that either have been addressed in the automatic authoring tools, or could be addressed with simple extensions.

Outside of the realm of automatic tutorial creation, future authoring tools would do well to include additional scaffolding to encourage users to include the types of information that I have described above, with some automatic formatting of tutorials post-authoring being possible. Examples of forms the scaffolding can take include separating and numbering steps, and including in each step a place for instructions and a place for explaining what the purpose of the step is. Websites that host tutorials could also use my collection of metrics as a basis to post guidelines for what should be included in tutorials posted on their sites.

4.5.3 Quality Differences Between Communities

My results suggest that you get what you pay for when it comes to tutorial authorship. PhotoshopTutorials, the site made up of professional tutorial authors, was the only one to have significantly higher quality than all of the other sites for a given metric, and this happened in five cases. PhotoshopTutorials also had significantly

higher values for quality than Google and Renderosity in seven quantitative metrics, and than Tutorialized in ten. Additionally, it had significantly higher than expected “Yes” counts in four of the categorical metrics. Finally, there was no metric in which PhotoshopTutorials performed significantly worse than any of the other sites.

The tutorials made by both PhotoshopTutorials, and Renderosity were longer, more complicated tutorials than those found in Tutorialized (or retrieved by Google Search using the CUTS technique). My personal observation was that those in PhotoshopTutorials in particular tended to focus on creating a full scene from many photographs. The complexity of these tutorials could be as a result of the authors having more invested in their creation. In the case of PhotoshopTutorials, the authors have several things at stake: they are getting paid to create the tutorial, and, more importantly, since they are being posted online with their names on them, these tutorials can represent their skill with Photoshop, and may serve as parts of their more formal portfolio. That Renderosity’s tutorials also seemed to have more effort put into them could be explained in the same way; tutorials on Renderosity are also associated with members’ online personae. The differences that do exist between the tutorials of these two sites may be explained by the fact that tutorials posted on PhotoshopTutorials are screened by the operators of the site, whereas Renderosity does not have any restrictions on what tutorials users can post.

4.6 Limitations

My study was an important starting point in the investigation of quality of online tutorials, but it is not without its limitations. These limitations include issues sur-

rounding generalization to wider populations, potentially missing factors relating to quality of tutorials, and a lack of first-hand examinations of users. I expand on these limitations below.

4.6.1 Generalization

My results likely can generalize from characterizing the quality of conventional text- and image-based online Photoshop tutorials to those of other large image editing applications, but my study only begun to scratch the surface in some related areas.

Video tutorials, for instance, while possessing some similarities to the more conventional tutorials that I focused on, also have some pretty significant differences. More investigation is needed to characterize the current quality of these tutorials, and to highlight areas where users would benefit from improved assistance.

Tutorials for applications in areas other than image editing are a similar story, although there is likely more generalizability than in the case of video tutorials. For example, many of the metrics that were uncovered in the literature were not domain-dependent, so those would likely be useful in future examinations of tutorials in other areas; however, my work stops short of being able to characterize existing tutorials outside the image editing domain. Some examples of domain-independent metrics include explaining why steps are necessary (*M16*), providing information on shortcuts (*M17*), and numbering steps (*M4*). As a contrasting example, previews of final work (*M21*) may not be applicable to all domains, particularly those whose output is less visual, such as in the domain of spreadsheet applications.

My work makes it clear that quality differences exist between different sources of

online tutorials, but because I only selected a single example of each community, more investigations are needed to test whether my findings generalize to all instances of a particular source. For instance, while PhotoshopTutorials outperformed the three other sites, I would hesitate to say that my findings suggest that in general tutorial factories will outperform the other communities. Verifying this generalization serves as a potential avenue for future work that may serve to better direct users to the best tutorials.

4.6.2 Missing Factors

While I investigated many factors relating to the quality of tutorials, there are likely some still missing. For instance, the list of metrics that I gathered is not exhaustive, as there likely are more indicators of quality that can be used. That being said, I was able to use the list to uncover quality differences between tutorials, indicating that the list is complete enough to function as a starting point in discussions of quality of tutorials.

Closely related to this, but with respect to metrics that websites use to showcase tutorials, that I didn't have any source with the social media buttons is a limitation; however, past studies of ratings indicate that single-point rating systems are less accurate than five-point rating systems [9]. As such, I would expect that the ratings that I examined in my study would be more accurate than the number of social media shares.

4.6.3 No User Study

My decision to study tutorials using metrics collected from previous work rather than run a user study has a few limitations. A user study could have given me first-hand access to people using tutorials, allowing me to see what problems they have, and providing me with the ability to conduct interviews with the users and consequently have a more clear picture of how they choose to rate tutorials. A user study also could have been used to determine and compare the importance of the different metrics. However, real users who are using tutorials have different motivations for following a tutorial than they would in a user study. The comments and ratings that I collected originate with these users who are reflecting on real-world experiences with the tutorial, so, while I was unable to explicitly ask them questions, I was privy to reflection from users in an ecologically valid context. Another benefit of my approach over a user study is that I was able to examine a wider variety of tutorials and give a more complete view of what is available; a user study by contrast would have required either a very large number of participants or very long sessions.

4.7 Summary

In this chapter I presented the study that I conducted for analyzing the quality of tutorials. This study involved collecting values for the metrics discussed in chapter 3 from Photoshop tutorials that came from four different online sources. The results of my study showed that the showcasing methods currently employed by tutorial websites, particularly ratings, and the number of times a tutorial is viewed, are failing

to properly express the range of quality in tutorials. I also found that while authors are frequently addressing many of the metrics that I collected, there is still some room for improvement; in particular, authors are often not explaining why steps are undertaken, annotating images, providing users with software version information and final-image previews, or numbering the steps of the tutorial. The different tutorial sources that I used showed that there is significant difference amongst online sources of tutorials, with PhotoshopTutorials, the source that uses tutorials authored by paid experts, significantly outperforming the other sites in many metrics, including having significantly better values than all three of the other sources in five of the quantitative metrics.

Chapter 5

Implementation Examples

In this chapter I present two example applications that I designed to make use of the findings from my study. In both cases, the applications have been implemented so far as was necessary to create the user interfaces. The first application is a tutorial authoring program; the second is a prototype design of a tutorial website. After describing the applications, I briefly outline some alternative ways of making use of my findings.

5.1 Tutorial Creator

The tutorial creator's main design intention is to provide scaffolding for authors to build new tutorials; scaffolding in this sense is a series of user interface components that remind authors of what information is helpful for users to complete the tutorial. To this aim, the design is made up primarily of a series of designated data entry areas. The application also integrates some tools to allow users to easily annotate images

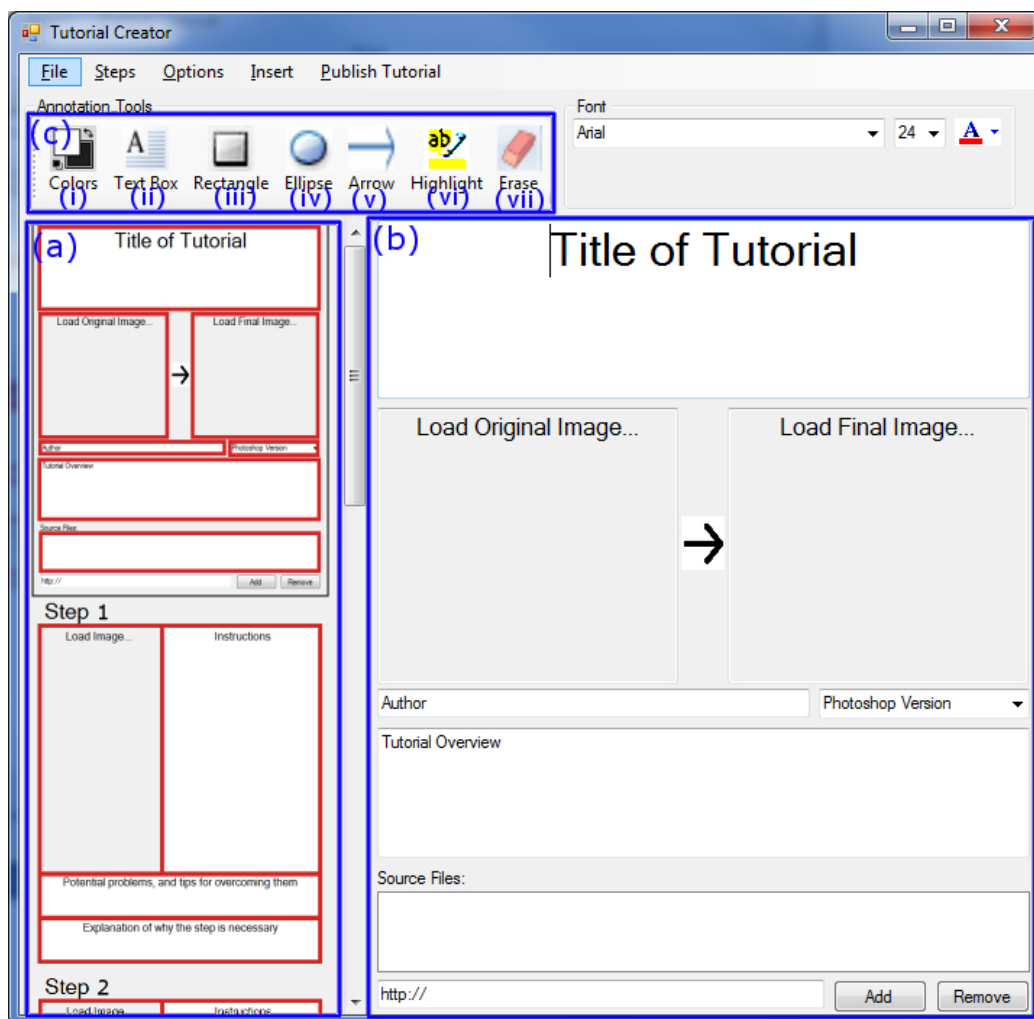


Figure 5.1: Tutorial creator application overview. (a) List of tiles. Allows users to select a tile to edit, and outlines missing content. (b) Main area for editing the content of the selected tile. (c) Tools for annotating images: (i) select colour for annotations; (ii–vi) add common annotation types; (vii) erase annotations.

in-place, and to quickly see what tutorial content is missing. Figure 5.1 shows the tutorial creator application with a brief explanation of the different components; I present a more thorough description of the components below.

Along the left side of the creator (Figure 5.1, (a)) is a list of tiles that serve several purposes. Each tile after the first is a miniaturized image of a step, while the

first tile is a miniaturized version of what I call the title tile, which is a collection of information that is frequently found at the start of tutorials. These tiles serve as buttons that load the full-size versions of the tile into the main editor pane (Figure 5.1, (b)), and red outlining on them indicates empty user-interface components, providing tutorial authors with a quick overview of what data is missing in the tutorial. Because annotations were underrepresented in many tutorials, I included a list of annotation tools along the top of the interface (Figure 5.1, (c)); they would be used to modify images within the tutorial by adding text, rectangles, circles, arrows, and highlighting.

The title tile contains interface components for specifying data that is associated with the tutorial on a whole, rather than any individual step. This includes text boxes for the title of the tutorial, the author's name, and a summary of the tutorial, picture boxes for the original and final images, a drop down for the version of Photoshop that was used, and a list for entering all of the web addresses for all of the source files that are needed to complete the tutorial. Figure 5.2 shows an example of a partially filled in title tile.

The scaffolding for each step addresses the things that steps generally contain, and explicitly encourages users to include the less commonly provided information as well. That is, each step contains a step number which is automatically generated, a picture box for an image, and text areas for instructions, tips for potential problems, and explanations of why the step is undertaken. Figure 5.3 shows a step tile that has had some of its information filled in.

To draw authors' attention to sections they've repeatedly left empty while not making users feel that they have to fill in every section for every step in order to

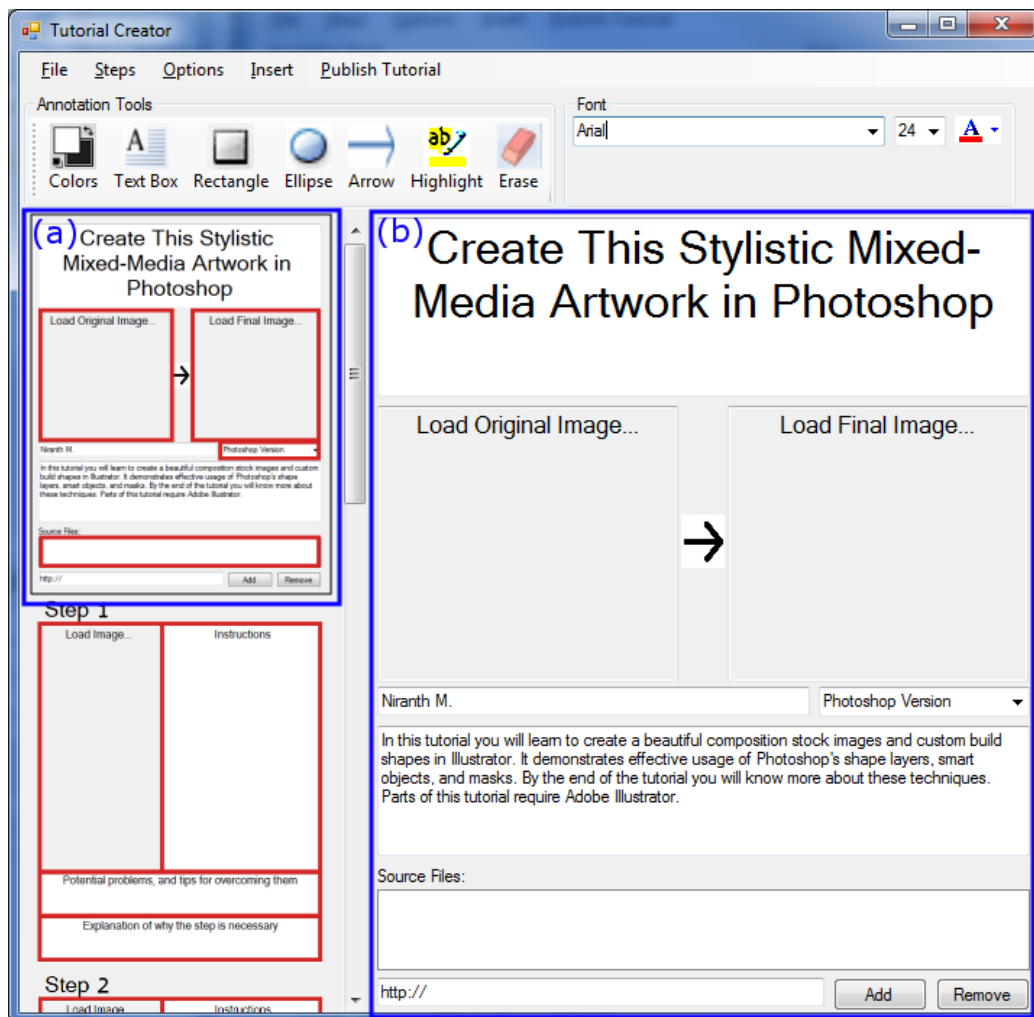


Figure 5.2: Editing the title tile in the tutorial creator application. (a) Overview of the title tile has red outlines to emphasize missing content. (b) The main editing area shows the title tile which allows users to enter common tutorial content, including the title, author's name, version of Photoshop used, overview of the tutorial, list of source files, and starting and ending images.

have a good tutorial, I vary the intensity of the red outlining that surrounds empty user-interface components based on the percentage of steps that also have that section empty.

In order to facilitate a clear explanation of what tools and menus are used in a step, the user is also presented with the ability to insert Photoshop tool and menu

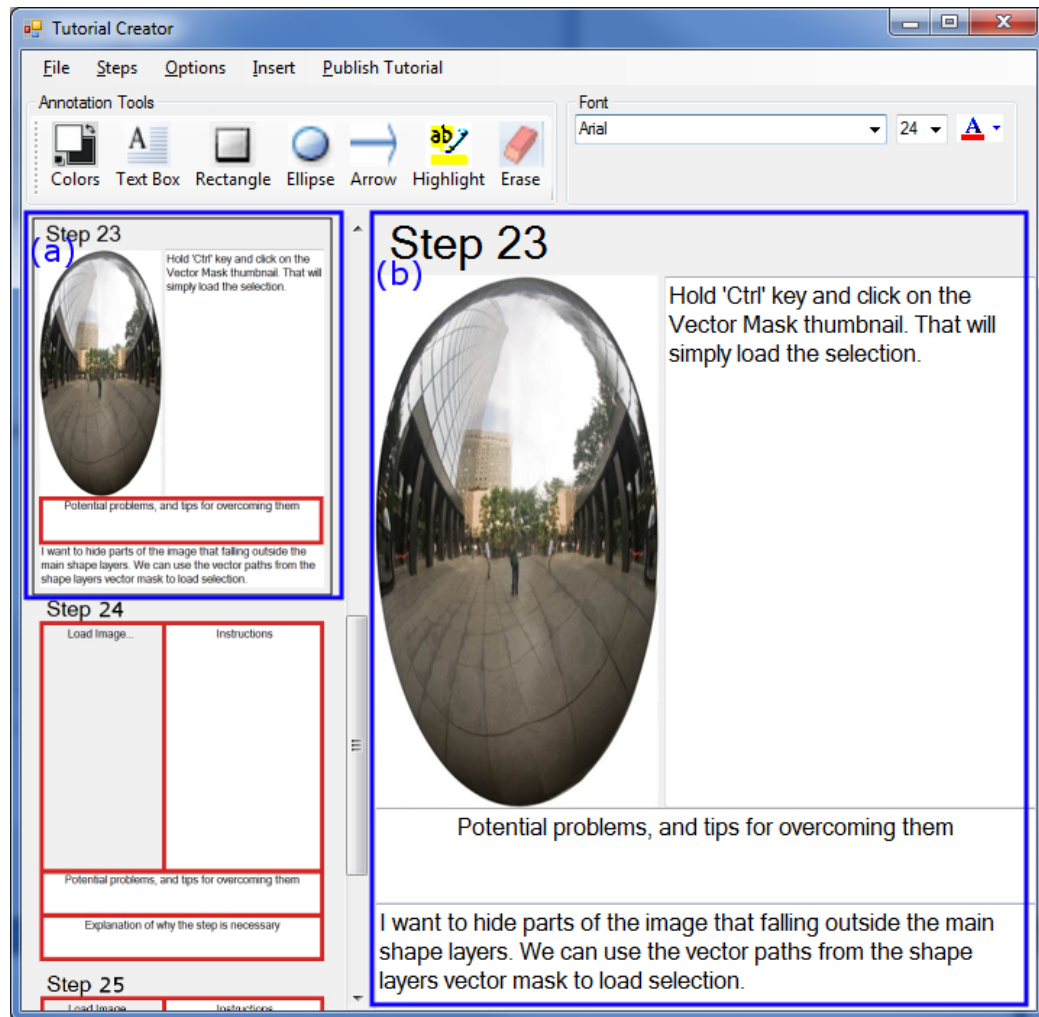


Figure 5.3: Editing a step in the tutorial creator application. (a) Overview of the step tile has red outlines to emphasize missing content. (b) The main editing window shows the selected step which has text boxes for entering instructions to complete the step, explanations of why the step is needed, and potential problems and tips for overcoming them. It also allows authors to upload and annotate an image to go along with the step.

references into the text. A menu item is selected by first selecting the *Insert* menu, followed by the *Photoshop Menu Reference* menu; the menu that opens up at this point is the same menu that the selected version of Photoshop includes, or if no version has been specified, an amalgamation of all menus from all the Photoshop

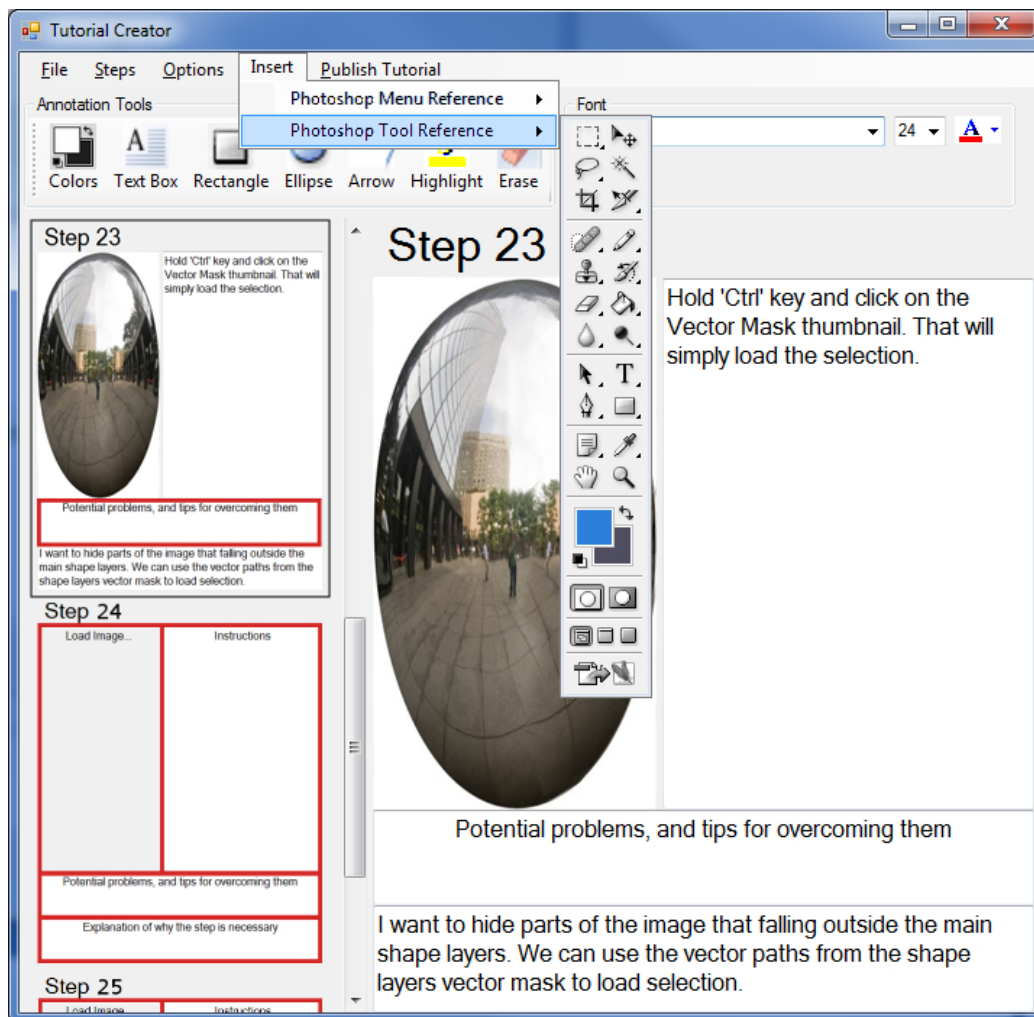


Figure 5.4: Inserting Photoshop tool references with the tutorial creator application. A menu that looks like the tool pallet from Photoshop appears. Selecting one of the tools enters the selected tool's name, representative thumbnail image, and keyboard shortcut, into the text of the creator.

versions. The tool reference is added in a similar way, but a special menu that looks like Photoshop's tool palette is shown from the *Photoshop Tool Reference* menu (see Figure 5.4). When a menu reference is inserted into text, it appears as the boldface text of each subsequent chosen menu item separated by arrows, with any available keyboard shortcuts in brackets following the list. For example, a reference to the

New... command from Photoshop's *File* menu looks like this: **File** → **New...** (**Ctrl+N**). This was a very common way of denoting menus and shortcuts that I came across in the tutorials. When a tool reference is inserted, it will insert the name of the tool, a thumbnail image of the tool as it appears in Photoshop's toolbox, and the shortcut for the tool in the same manner as was done for the menu shortcuts. The thumbnail image helps users identify tools that they may not recognize from the name alone.

5.2 Tutorial Website

The tutorial website example that I constructed has a number of features based on the findings of my study. Figure 5.5 shows the main page for the website, and Figure 5.6 shows a selected tutorial. Below, I discuss the features included in the images.

In order to provide users with some numeric guidance on the quality of the tutorials, I broke the ratings up into several different categories, taking advantage of the idea that quality is a multifaceted property. This also builds on results from the studies on ratings that Cosley et al. [9] conducted using their Movie Lens application. They found that users prefer fine-grained rating scales because they allow users to express their opinions more completely; providing users with multiple categories to rate should allow them to more completely share their opinions than any overall rating would. I focused on user comments for determining what rating categories I would use because the comments provide insight into what the users are thinking when they finish reading, or attempting to follow, a tutorial; these thoughts likely



Figure 5.5: Selecting a tutorial in the tutorial website. (a) Each tutorial is listed as a set of basic tutorial information: title, author, date posted, final image, and short introduction. (b) Four rating categories for each tutorial are visible, with positive and negative values for each: (i) helpfulness; (ii) coolness; (iii) instructional ability; and (iv) ease of following. (c) Tooltips for each rating display the type of rating, an explanation of what the rating means, and a breakdown of the rating by the raters' experience with Photoshop. (d) The tutorials can be sorted based on a combination of rating values and other tutorial data.

impacted what the users rated the tutorial.

Comments left by users frequently stated how much they liked the effects depicted in tutorials, often while simultaneously asking for help with a step; to this end I selected (1) *helpfulness* and (2) *coolness* as two rating categories. The next rating category that I selected, (3) *instructional ability*, was based on frequent praise in comments for authors' ability to teach new skills, and make complicated steps easy to understand. Readers also often would comment on becoming lost at certain points in a tutorial; this lead to me selecting (4) *ease of following* as my final category. I use icons to represent each category when they are presented.

Because users are more likely to rate numerous rating values if it is simple to do so, and finer grained scales take longer to rate [57], I use the binary thumbs up and thumbs down rating scale. In an effort to further combat the lack of clarity that appears to have been a problem in the original ratings, I also include tooltips on the icons to provide users with guidance for selecting a given rating; these tooltips are phrased as "I" statements that can be agreed (thumbs up) or disagreed with (thumbs down). One such tooltip is shown in Figure 5.6 (d). Below, I list each of the "I" statements.

1. **Helpfulness:** I think the author makes adequate attempts at helping users to avoid potential problems.
2. **Coolness:** I think the effects depicted in the tutorial are awesome.
3. **Instructional Ability:** I learned a new technique through reading or following this tutorial.

4. Ease of Following: I found that the instructions were clear and easy to follow.

Because a reader's proficiency with Photoshop is likely to impact how well they follow, and what they're looking for, in a tutorial, the tooltips on the tutorial search page (Figure 5.5, (c)) also include a breakdown of the ratings based on levels of expertise. These levels are intended to be self-reported, and associated with log-in ID's. I used three levels for expertise (amateur, intermediate, and professional), but determining the best number of levels for describing expertise is outside of the scope of this thesis.

The typical way for tutorials to be sorted is to choose one attribute, either belonging to the tutorial (date uploaded, title, and author), or calculated as a result of users' input (rating, views, and votes) on which to sort. This method ignores the idea that quality is multidimensional. Instead, my implementation allows users to choose an order amongst the different rating categories and other tutorial information (Figure 5.5, (d)) on which to sort the tutorials.

Once a tutorial is selected, the user can choose to either "thumb up" or "thumb down" any of the four rating categories (Figure 5.6, (c)). The rating counts will not be displayed on this page because viewing ratings for an item can affect how a user chooses to rate that item [9]. This effect cannot be entirely avoided, however, because the user has already had a chance to see the tutorial's ratings, and potentially chose to view it based on them.



Figure 5.6: Viewing a tutorial in the tutorial website. (a) The tutorial is visible in the main viewing panel. (b) Comments are visible at all times. (c) The tutorial can be up- or down-rated in the four rating categories; rating totals are intentionally not visible. (d) Tooltips for each rating display the type of rating and an explanation of what it means.

5.3 Alternative Implementations

The implementations that I have included above take advantage of many of the findings of my thesis work, but they are not the only way to apply them. In this section I outline some alternative methods for implementing my results.

Automatic tutorial authoring systems (such as those created by Chi et al. [8] and Grabler et al. [21]) could be augmented to also collect resource lists, version information, and any of the other sets of information that I have highlighted as being commonly left out. Alternatively, a system could use a combination of automation and scaffolding to collect what information it can automatically, but still remind authors that information that is difficult to automatically collect, such as why a step is performed or tips for overcoming potential problems, can also be helpful for the readers.

Some more ways of drawing an author's attention to areas of the tutorial that are in need of attention could also be included in alternative systems. Examples of this include raising warning messages when the author attempts to publish a tutorial that contains a high percentage of empty scaffolding. A system could also attempt to match things that are mentioned in text to things that are present in images and suggest automatic annotations. For example, when a user enters a reference to a tool, if the tool palette is found to be present in the step's image, then it could suggest highlighting that tool in the image.

Due to the lack of accuracy discovered in the single dimension rating system, I believe it is beneficial to explore alternative systems. To this end, I included a rating system that uses multiple dimensions to gather information from the user on different

aspects of quality, but the number of things that define quality are too numerous to include everything. I chose a set of four dimensions that represented many of the aspects of quality that users address in comments, but user studies will be needed to verify that this choice of categories adequately describes the quality of the tutorials; this is a good direction for future work.

Another consideration in defining a rating system is how to collect the ratings from the users. I chose to use the thumbs-rating system because I would expect it to be less likely that users are going to assign ratings to multiple categories than to only a single category; Cosley et al. [9] showed that what we gain from using a less precise rating is less mental effort, less time required to rate the tutorials, and an increased likelihood of the user choosing to submit a rating. However, this can result in users feeling that the rating is less accurate. An other option that would still take advantage of the multifaceted aspect of quality, would be to use a set of 5-point ratings. This would be expected to give more accurate ratings, but with a lower chance of them actually being filled in.

An alternative method of rating determination could be to calculate a set of scores based on the values of all of the metrics discussed in chapter 3. A drawback of this type of system is that automating this process could be quite difficult for many of the metrics. This could leave manual collection as the only option for gathering metric values, which would need to be conducted by a staff member of the website it was uploaded to. The time manual collection of the metric values takes would likely not be worth the increase in accuracy to a website owner. That being said, how to automatically collect these values, and an examination of how to combine them into

a set of ratings, would also make an interesting avenue for future work.

I included a form of sorting that allows users to sort based on a combination of factors, including the ratings. This sorting emphasizes that users care about multiple factors, and gives the individual user input on what factors they personally deem to be important, but a more elaborate system could easily be implemented that would allow users to exclude tutorials based on criteria such as minimum rating values and date ranges.

5.4 Summary

In this chapter, I presented two example applications that demonstrate how to apply my findings in the real world. The first was a tutorial authoring tool that provides scaffolding for important information, and encourages authors to fill in under-represented details. The second was a tutorial presentation site that implements a multidimensional rating system intended to address the inaccuracy that I found plagues Tutorialized's rating system.

Chapter 6

Conclusion

6.1 Summary

In my thesis, I reported on the current state of the quality of end-user-authored online Photoshop tutorials, and presented designs for two applications that address the primary limitations I uncovered in the tutorials. I summarize this work below by addressing the three primary research questions that I posed in the introduction.

1. What is the quality of the tutorials that are available for users on the web?

In order to answer this question, I used user comments and existing literature to build a set of 24 metrics that could be used to describe the quality of tutorials, and then applied these metrics to 154 text- and image-based Photoshop tutorials.

My examination of the metrics revealed that while most of the guidelines for creating effective tutorials have been followed with substantial regularity, there were

a few that had been largely ignored by the tutorial authoring community. The ignored metrics represent lost opportunities in teaching use of the application.

2. Does the quality of tutorials vary according to characteristics of the authoring community?

The tutorials that I applied the metrics to were sampled from four sources in order to ascertain whether a tutorial's authoring community can have a predictive effect on its quality.

I found that the quality of tutorials was consistently higher for those that were posted on the website that paid Photoshop experts to author them. This implies that typical authors may benefit from assistance in creating tutorials that follow the common practices of experts.

3. Are currently used mechanisms for highlighting tutorial quality effectively distinguishing between the range of tutorials available?

I addressed this question by collecting values for several showcasing strategies from all of the tutorials available from the tutorial aggregator source.

My analysis revealed that the current methods of displaying information about tutorial quality are of questionable use. The most direct method of advertising a tutorial's quality, the average rating, barely varied at all throughout the sample, despite substantial evidence that quality does vary. Other prominent showcasing methods have a great number of factors not related to quality that impact them. For

example, the number of times that a tutorial has been viewed is tied to the length of time that a tutorial has been on a site, the level of traffic on the site during that time, the tutorial's average rating, and the interest in the topic of the tutorial.

6.2 Future Work

My research functioned as a starting point in the exploration of the quality of online tutorials, and consequently has exposed several areas for future research. These areas include validating the set of metrics that I used, expanding my work to video tutorials, further investigating ratings' ability to showcase tutorials' quality, and the potential for automatic rating calculations.

An important avenue of future research is to formally validate my proposed set of quality metrics. One possible way of doing this would be to run a user study that has users attempt unfamiliar tasks with Photoshop, relying on tutorials to teach them how to complete them. To accomplish this levels of each metric would need to be defined (e.g., how many steps count as low, medium, and, high numbers). Tutorials would then need to be selected in such a way as to span the assigned levels of each metric. We could then validate the metrics by comparing users' completion times and errors across the different levels. In order to test how well the user retained the knowledge, an important aspect of the quality of a tutorial, the user would need to complete a different task that uses the same skill set at a later date. The challenge in running such a study is that it would require that users complete a large number of tutorials.

Another interesting area of pursuit that arises from my work is determining a

similar set of metrics for video tutorials. Many of the metrics that I have proposed likely still apply (e.g., the number of steps), but others are less easy to translate to this domain (e.g., number of images). A review of user comments and previous works, but with a focus on how the guidelines apply to video tutorials, may also prove fruitful in this domain.

My finding that ratings do not accurately expose the quality of the tutorials raises several interesting questions, including: i) Does this finding extend to other Photoshop tutorial sites that have five-star ratings? ii) Does it also extend to tutorial sites for other applications? iii) Do multidimensional ratings lead to a more accurate assessment of quality? And, if so, iv) What scales should we use?

Similarly, it would be interesting to explore whether social-media button clicks, the showcasing metric that was unfortunately not provided by any of my tutorial sources, provide more accurate representations of the tutorials' quality than the five-star ratings do.

My work highlights metrics which may be possible to use to generate more accurate ratings than the ones currently collected from users. This idea relies on creating a system that automates collection of the metric values. Then, it may be able to attribute an overall rating, or a set of ratings, by comparing the values assigned to one tutorial's metrics with those assigned to the others. While automatic collection would be an easy task for some metrics, others would be more challenging. These values may be easier to collect by ensuring that tutorials are authored in a tutorial application that uses a technique, such as scaffolding, to identify them. Another challenge of this work is determining how to combine the values of the metrics into a

rating or set of ratings in a meaningful way.

Bibliography

- [1] ADOMAVICIUS, G., AND KWON, Y. New recommendation techniques for multi-criteria rating systems. *IEEE Intelligent Systems* 22, 3 (May/June 2007), 48–55.
- [2] AGRAWALA, M., PHAN, D., HEISER, J., HAYMAKER, J., KLINGNER, J., HANRAHAN, P., AND TVERSKY, B. Designing effective step-by-step assembly instructions. *ACM Transactions on Graphics* 22, 3 (July 2003), 828–837.
- [3] ASCH, S. E. Effects of group pressure upon the modification and distortion of judgments. In *Groups, Leadership and Men: Research in Human Relations*, H. Guetzkow, Ed. Carnegie Press, Oxford, England, 1951, pp. 177–190.
- [4] BOOHER, H. R. Relative comprehensibility of pictorial information and printed words in proceduralized instructions. *Human Factors* 17, 3 (June 1975), 266–277.
- [5] BREESE, J. S., HECKERMAN, D., AND KADIE, C. Emperical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI)* (July 1998), Morgan Kaufmann, San Francisco, CA, USA, pp. 43–52.

-
- [6] BUNT, A., CONATI, C., AND MCGRENERE, J. Supporting interface customization using a mixed-initiative approach. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)* (January 2007), ACM, pp. 92–101.
 - [7] CARROLL, J. *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. The Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA, 1990.
 - [8] CHI, P.-Y., AHN, S., REN, A., DONTCHEVA, M., LI, W., AND HARTMANN, B. MixT: Automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST)* (October 2012), ACM, pp. 93–102.
 - [9] COSLEY, D., LAM, S. K., ALBERT, I., KONSTAN, J. A., AND RIEDL, J. Is seeing believing? How recommender interfaces affect users opinions. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2003), ACM, pp. 585–592.
 - [10] DENNING, J. D., KERR, W. B., AND PELLACINI, F. MeshFlow: Interactive visualization of mesh construction sequences. *ACM Transactions on Graphics* 30, 4 (July 2011), 66:1–66:8.
 - [11] EHRET, B. D. Learning where to look: Location learning in graphical user interfaces. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2002), ACM, pp. 211–218.

-
- [12] EKSTRAND, M., LI, W., GROSSMAN, T., MATEJKA, J., AND FITZMAURICE, G. Searching for software learning resources using application context. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST)* (October 2011), ACM, pp. 195–204.
 - [13] FARKAS, D. K. The role of balloon help. *Journal of Computer Documentation* 17, 2 (May 1993), 3–19.
 - [14] FARKAS, D. K., AND WILLIAMS, T. R. John Carroll’s ‘The Nurnberg Funnel’ and minimalist documentaions. *IEEE Transactions on Professional Communication* 33, 4 (December 1990), 182–187.
 - [15] FENG, G., ZHIYOU, L., AND HUIXIN, L. Collaborative filtering algorithm based on the preference list in the digital library. In *Information Computing and Applications* (September 2012), vol. 307 of *Communications in Computer Science*, Springer, pp. 448–455.
 - [16] FINDLATER, L., AND MCGRENERE, J. A comparison of static, adaptive, and adaptable menus. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2004), ACM, pp. 89–96.
 - [17] FOURNEY, A., LAFRENIERE, B., MANN, R., AND TERRY, M. ‘Then click OK!’: Extracting references to interface elements in online documentaion. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (May 2012), ACM, pp. 35–38.

-
- [18] FOURNEY, A., MANN, R., AND TERRY, M. Characterizing the usability of interactive applications through query log analysis. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (May 2011), ACM, pp. 1817–1826.
 - [19] GAJOS, K. Z., CZERWINSKI, M., TAN, D. S., AND WELD, D. S. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)* (May 2006), ACM, pp. 201–208.
 - [20] GAJOS, K. Z., EVERITT, K., TAN, D. S., CZERWINSKI, M., AND WELD, D. S. Predictability and accuracy in adaptive user interfaces. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2008), ACM, pp. 1271–1274.
 - [21] GRABLER, F., AGRAWALA, M., LI, W., DONTCHEVA, M., AND IGARASHI, T. Generating photo manipulation tutorials by demonstration. *ACM Transactions on Graphics* 28, 3 (August 2009), 66:1–66:9.
 - [22] GREENBERG, S., AND WITTEN, I. H. Adaptive personalized interfaces: A question of viability. *Behaviour and Information Technology* 4, 1 (January 1985), 31–45.
 - [23] GROSSMAN, T., AND FITZMAURICE, G. ToolClips: An investigation of contextual video assistance for functionality understanding. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2010), ACM, pp. 1515–1524.

-
- [24] GROSSMAN, T., FITZMAURICE, G., AND ATTAR, R. A survey of software learnability: Metrics, methodologies and guidelines. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2009), ACM, pp. 649–658.
- [25] GROSSMAN, T., MATEJKA, J., AND FITZMAURICE, G. Chronicle: Capture, exploration, and playback of document workflow histories. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST)* (October 2010), ACM, pp. 143–152.
- [26] HEISER, J., PHAN, D., AGRAWALA, M., TVERSKY, B., AND HANRAHAN, P. Identification and validation of cognitive design principles for automated generation of assembly instructions. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)* (May 2004), ACM, pp. 311–319.
- [27] HORVITZ, E., BREESE, J., HECKERMAN, D., HOVEL, D., AND ROMMELSE, K. The lumire project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI)* (July 1998), Morgan Kaufmann, San Francisco, CA, USA, pp. 256–265.
- [28] KELLEHER, C., AND PAUSCH, R. Stencils-based tutorials: Design and evaluation. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2005), ACM, pp. 541–550.

-
- [29] KNABE, K. Apple guide: A case study in user-aided design of online help. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (May 1995), ACM, pp. 286–287.
- [30] KONG, N., GROSSMAN, T., HARTMANN, B., FITZMAURICE, G., AND AGRAWALA, M. Delta: A tool for representing and comparing workflows. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (May 2012), ACM, pp. 1027–1036.
- [31] KONG, W., LIU, Q., YANG, Z., AND HAN, S. Collaborative filtering algorithm incorporated with cluster-based expert selection. *Journal of Information and Computational Science* 9, 12 (2012), 3421–3429.
- [32] LAFRENIERE, B., BUNT, A., LOUNT, M., KRYNICKI, F., AND TERRY, M. A. AdaptableGIMP: Designing a socially-adaptable interface. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST)* (October 2011), ACM, pp. 89–90.
- [33] LAFRENIERE, B., BUNT, A., LOUNT, M., AND TERRY, M. Understanding the roles and uses of web tutorials. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)* (July 2013), AAAI, Palo Alto, CA, USA, pp. 303–310.
- [34] LAFRENIERE, B., GROSSMAN, T., AND FITZMAURICE, G. Community enhanced tutorials: Improving tutorials with multiple demonstrations. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2013), ACM.

-
- [35] LAPUT, G., ADAR, E., DONTCHEVA, M., AND LI, W. Tutorial-based interfaces for cloud-enabled applications. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST)* (October 2012), ACM, pp. 113–122.
- [36] LAU, T., DREW, C., AND NICHOLS, J. Interpreting written how-to instructions. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)* (July 2009), Morgan Kaufmann, San Francisco, CA, USA, pp. 1433–1438.
- [37] LEE, H.-H., AND TENG, W.-G. Incorporating multi-criteria ratings in recommendation systems. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI)* (August 2007), IEEE, pp. 273–278.
- [38] LINTON, F., JOY, D., SHAEFER, H.-P., AND CHARRON, A. OWL: A recommender system for organization-wide learning. *Journal of Educational Technology and Society* 3, 1 (January 2000), 62–76.
- [39] LINTON, F., AND SHAEFER, H.-P. Recommender systems for learning: Building user and expert models through long-term observation of application use. *User Modeling and User-Adapted Interaction* 10, 2-3 (June 2000), 181–207.
- [40] LOUNT, M., BUNT, A., LAFRENIERE, B., KRYNICKI, F., AND TERRY, M. A. AdaptableGIMP: A socially adaptable interface. Poster Presentation at Second Annual Conference of Graphics, Animation, and New Media (GRAND), May 2011.

-
- [41] LOUNT, M., BUNT, A., LAFRENIERE, B., AND TERRY, M. A. Intelligent support for community task-based personalization. Poster Presentation at First Annual Conference of Graphics, Animation, and New Media (GRAND), June 2010.
- [42] MATEJKA, J., GROSSMAN, T., AND FITZMAURICE, G. Ambient help. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (May 2011), ACM.
- [43] MATEJKA, J., LI, W., GROSSMAN, T., AND FITZMAUTICE, G. CommunityCommands: Command recommendations for software applications. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST)* (October 2009), ACM, pp. 193–202.
- [44] MCGRENERE, J., BAECKER, R. M., AND BOOTH, K. S. An evaluation of a multiple interface design solution for bloated software. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 2002), ACM.
- [45] MCGRENERE, J., AND MOORE, G. Are we all in the same "bloat"? In *Proceedings of Graphics Interface (GI)* (May 2000), Morgan Kaufmann, San Francisco, CA, USA, pp. 187–196.
- [46] MITCHELL, J., AND SHNEIDERMAN, B. Dynamic versus static menus: An exploratory comparison. *ACM SIGCHI Bulletin* 20, 4 (April 1989), 33–37.

-
- [47] NASS, C., AND MOON, Y. Machines and mindlessness: Social responses to computers. *Journal of Social Issues* 56, 1 (Spring 2000), 81–103.
- [48] NOVICK, L. R., AND MORSE, D. L. Folding a fish, making a mushroom: The role of diagrams in executing assembly procedures. *Memory and Cognition* 28, 7 (October 2000), 1242–1256.
- [49] OPPERMAN, R., AND SIMM, H. *Adaptability: User-Initiated Individualism*. Lawrence Erlbaum Associates, 1994, ch. 1, pp. 54–63.
- [50] PALMITER, S., AND ELKERTON, J. An evaluation of animated demonstrations of learning computer-based tasks. In *Proceedings of the Annual ACM SIGCHI Conference on Human Factors in Computing System (CHI)* (April 1991), ACM, pp. 253–263.
- [51] PALMITER, S., AND ELKERTON, J. Animated demonstrations for learning procedural computer-based tasks. *Human-Computer Interaction* 8, 3 (September 1993), 193–216.
- [52] PARK, J., HAN, S. H., PARK, Y. S., AND CHO, Y. Adaptable versus adaptive menus on the desktop: Performance and user satisfaction. *International Journal of Industrial Ergonomics* 37, 8 (August 2007), 675–684.
- [53] PENNOCK, D. M., HORVITZ, E., LAWRENCE, S., AND GILES, C. L. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proceedings of the 16th Conference on Uncertainty in Artificial*

- Intelligence (UAI)* (June 2000), Morgan Kaufmann, San Francisco, CA, USA, pp. 473–480.
- [54] PONGNUMKUL, S., DONTCHEVA, M., LI, W., WANG, J., BOURDEV, L., AVIDAN, S., AND COHEN, M. F. Pause-and-Play: Automatically linking screencast video tutorials with applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST)* (October 2011), ACM, pp. 135–144.
- [55] SAHOO, N., KRISHNAN, R., DUNCAN, G., AND CALLAN, J. The halo effect in multicomponent ratings and its implications for recommender systems: The case of Yahoo! movies. *Information Systems Research* 23, 1 (March 2012), 231–246.
- [56] SEARS, A., AND SHNEIDERMAN, B. Split menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction (TOCHI)* 1, 2 (March 1994), 27–51.
- [57] SPARLING, E. I., AND SEN, S. Rating: How difficult is it? In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys)* (Chicago, IL, October 2011), ACM, pp. 149–156.
- [58] STERN, K. R. An evaluation of written, graphics, and voice messages in proceduralized instructions. In *Proceedings of the Human Factors and Ergonomics Society 28th Annual Meeting (HFES)* (1984), vol. 28, HFES, Santa Monica, CA, USA, pp. 314–318.

-
- [59] STUERZLINGER, W., CHAPUIS, O., PHILLIPS, D., AND ROUSSEL, N. User interface faades: Towards fully adaptable user interfaces. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST)* (October 2006), ACM, pp. 309–318.
- [60] TUKEY, J. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977.
- [61] WEI, S., YE, N., ZHANG, S., HUANG, X., AND ZHU, J. Item-based collaborative filtering recommendation algorithm combining item category with interestingness measure. In *Proceedings of the International Conference on Computer Science and Service System (CSSS)* (August 2012), IEEE, pp. 2038–2041.