BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

IMPACT OF SUBJECTIVE VISUAL PERCEPTION ON AUTOMATIC EVALUATION OF DASHBOARD DESIGN

VLIV SUBJEKTIVNÍHO VIZUÁLNÍHO VNÍMÁNÍ NA AUTOMATICKÉ HODNOCENÍ VZHLEDU ROZHRANÍ DASHBOARD

DOCTORAL THESIS SUMMARY
AUTOREFERÁT DISERTAČNÍ PRÁCE

AUTHOR — **Ing. JIŘÍ HYNEK**
AUTOR PRÁCE

SUPERVISOR — **prof. Ing. TOMÁŠ HRUŠKA, CSc.**
ŠKOLITEL

BRNO 2019

Abstract

Using metrics and quantitative design guidelines to analyze design aspects of user interfaces (UI) seems to be a promising way for the automatic evaluation of the visual quality of user interfaces. Since this approach is not able to replace user testing, it can provide additional information about possible design problems in early design phases and save time and expenses in the future. Analyses of used colors or UI layout are the examples of such evaluation. UI designers can use known pixel-based (e.g., *Colorfulness*) or object-based (e.g., *Balance* or *Symmetry*) metrics which measure chosen UI characteristics, based on the raster or structural representation of UI.

The problem of the metric-based approach is that it does not usually consider users' subjective perception (e.g., subjective perception of color and graphical elements located on a screen). Today's user interfaces (e.g., dashboards) are complex. They consist of several color layers, contain overlapping graphical elements, which might increase ambiguity of users' perception. It might be complicated to select graphical elements for the metric-based analysis of UI, so the selection reflects users' perception and principles of a visual grouping of the perceived shapes (as described by Gestalt psychology). Development of objective metrics and design guidelines usually requires a sufficiently large training set of user interface samples annotated by a sufficient number of users.

This thesis focuses on the automatic evaluation of dashboard design. It combines common knowledge about dashboards with the findings in the field of data visualization, visual perception and user interface evaluation, and explores the idea of the automatic evaluation of dashboard design using the metric-based approach. It analyzes chosen pixel-based and object-based metrics. It gathers the experience of users manually segmenting dashboard screens and uses the knowledge in order to analyze the ability of the object-based metrics to distinguish well-designed dashboards objectively. It establishes a framework for the design and improvement of metrics and proposes an improvement of selected metrics. It designs a new method for segmentation of dashboards into regions which are used as inputs for object-based metrics. Finally, it compares selected metrics with user reviews and asks questions suggesting future research tasks.

Keywords

dashboard, user interface, UX, usability evaluation, metrics, aesthetics, balance, visual perception, subjective perception, Gestalt laws, segmentation

Abstrakt

Analýza vlastností uživatelských rozhraní založená na použití metrik a kvantitativních pravidel grafického designu se zdá být slibným přístupem pro automatické hodnocení vizuální kvality uživatelských rozhraní. Přestože tento přístup nemůže plně nahradit uživatelské testování, může poskytnout dodatečné informace o možných problémech návrhu uživatelských rozhraní v počátečních fázích vývoje a ušetřit tím čas a výdaje v budoucnu. Příkladem je analýza použitých barev a rozvržení grafických elementů na obrazovce. Návrháři uživatelských rozhraní mohou měřit vlastnosti uživatelských rozhraní za použití známých metrik založených na analýze pixelů bitmapy (např. barevnost) nebo grafických elementů (např. vyvážení, symetrie).

Problémem použití metrik nicméně je, že tento přístup zpravidla nezohledňuje subjektivní vnímání uživatelů (např. subjektivní vnímání barev nebo grafických elementů rozmístěných na obrazovce). Dnešní uživatelská rozhraní (jako například rozhraní dashboard) jsou komplexní. Skládají se z několika barevných vrstev, obsahují překrývající se grafické elementy, což může zvyšovat nejednoznačnost vnímání tohoto rozhraní uživateli. Může být proto komplikované vybrat takové grafické elementy, které odpovídají elementům rozpoznaným uživateli v souvislosti s principy shlukování vnímaných tvarů (jak je popsáno Gestalt psychologií). Vývoj objektivních metrik a kvantitativních pravidel grafického designu obvykle vyžaduje dostatečně velkou trénovací množinu vzorků uživatelských rozhraní anotovaných dostatečným počtem uživatelů.

Tato práce se zaobírá automatickým ověřováním vzhledu uživatelských rozhraní dashboard. Práce kombinuje obecné znalosti týkající rozhraní dashboard s poznatky z oblasti vizualizace dat, vizuálního vnímání a ověřování kvality uživatelských rozhraní a následně zkoumá myšlenku automatického hodnocení vzhledu rozhraní dashboard s využitím metrik. Práce analyzuje vybrané metriky založené na analýze pixelů bitmapy a grafických elementů. Konkrétně zkoumá, jakým způsobem uživatelé rozpoznávají grafické elementy v rozhraních dashboard a výsledky aplikuje pro hodnocení schopnosti metrik (analyzujících grafické elementy rozhraní) objektivně rozpoznávat dobře navržené vzorky rozhraní dashboard. Dále představuje framework pro návrh a vylepšení metrik, který využívá pro vylepšení vybraných metrik. Následně navrhuje metodu pro segmentaci rozhraní dashboard do regionů, které mohou být použity jako vstupy pro tyto metriky. Závěrem práce porovnává vybrané metriky s hodnocením rozhraní uživateli a pokládá otázky vhodné pro budoucí výzkum.

Klíčová slova

dashboard, uživatelské rozhraní, UX, testování použitelnosti, metriky, estetičnost, vyvážení, vizuální vnímání, subjektivní vnímání, Gestalt principy, segmentace

Impact of Subjective Visual Perception on Automatic Evaluation of Dashboard Design

Declaration

Hereby I declare that this Ph.D. thesis was prepared as an original author's work under the supervision of prof. Ing. Tomáš Hruška CSc. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

....
Jiří Hynek
June 24, 2019

Acknowledgements

First of all, I would like to thank to my supervisor prof. Ing. Tomáš Hruška CSc. who kept supervising and consulting the topic of my thesis for all the years of the study. Then, I would like to thank to my colleagues at Brno University of Technology, especially at Department of Information Systems, Faculty of Information Technology. I thank to (in alphabetic order) Ing. Radek Burget Ph.D., doc. RNDr. Jitka Kreslíková, CSc., and RNDr. Libor Škarvada for advice and consultations. I thank to Ing. Olena Pastushenko for cooperation on the selected publications. I thank to all students who participated in the studies and experiments, helped me improve the software used in the thesis or who somehow cooperated with me. Last, but not least, I would like to thank to my family, especially to my parents without whose support I wouldn't be able to finish the thesis. I also thank to friends and all people who supported me to finish the thesis.

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project 'IT4Innovations excellence in science – LQ1602'.

Reference

HYNEK, Jiří. *Impact of Subjective Visual Perception on Automatic Evaluation of Dashboard Design*. Brno, 2019. Doctoral thesis summary. Brno University of Technology, Faculty of Information Technology. Supervisor prof. Ing. Tomáš Hruška, CSc.

Contents

Chapter 1

Introduction

Dashboard is a frequently used term connected with business intelligence and management information systems. It is a favorite tool used by many organizations to comprehensively present their data for operational, analytical, or strategic purposes. It presents key performance indicators which help to evaluate the progress and benefit of business activities [Eckerson, 2006]. Since dashboards support decision-making, they have become popular among a wide range of users for the management of personal activities. We can find numerous web applications providing dashboard templates to visualize data gathered from common services like social networks. The rising diversity of dashboards has led UI designers and researchers to think about the principles of high-quality dashboard design.

One of the first rules which brought some clarity to dashboard characteristics were provided by Stephen Few [2006]. He has worked with the idea of a single screen display comprehensively presenting the most critical information to achieve goals. The requirement of the dashboard—"present information on a single screen"—is what distinguishes dashboards from other interfaces and, also, makes them difficult to design. UI designers need to focus on the design aspects such as strong simplification, elimination of unnecessary elements, highlighting significant relationships between data, or careful selection of graphical elements capable of comprehensively presenting a great deal of data using a small area. Few pointed out that most of the existing so-called dashboards break the requirement. He has provided a framework based on the knowledge of famous books regarding design and graphics (e.g., [Tufte, 2001; Ware, 2004]). This framework contains heuristics for the dashboard design, including examples of well-designed dashboards.

Even more than ten years after the release of Few's publication, we can still observe that the majority of dashboards ignore Few's heuristics or express them in their own way. We assume that the reason might be the complexity and vague definition of the framework and the lack of other sources which would provide formal and quantitative knowledge in the area of dashboard design. For instance, the selection of appropriate charts and colors usually depends on an actual context, and it cannot be completely generalized. A dashboard designer needs to be a person with experience in human-computer interaction and capable of applying the framework correctly. The presence of users is usually required to evaluate usability, which increases the time and expenses of the design phase.

A challenge in improving UI design and evaluation is that of finding quantitative guidelines which would detect some of the design problems and help to distinguish well-designed interfaces from poorly designed ones. Such guidelines could be applied automatically during the early design phase without the presence of users and specialists in UI design [Ivory and Hearst, 2001]. The simplicity of guidelines is, however, the major weakness of this approach. It is not usually easy to describe complex design attributes of a user interface since they usually depend on the subjective judgment of the viewer. Design guidelines are usually based on simple metrics (e.g., the average colorfulness based on saturation of screen pixels [Reinecke et al., 2013]).

One possible step in making the metric-based evaluation more reliable is to process a screen similarly as it is perceived by the human brain—not as a matrix of pixels but as a group of objects within a scene as described (for example) by Baker et al. [2009]. Then, we evaluate objects on a screen (e.g., controls and widgets) and their properties (e.g., size or position) as described by Charfi et al. [2014]. For this purpose, we use *object-based* metrics. We can measure advanced characteristics of a screen (e.g., the characteristics connected with layouts). For instance, Ngo et al. [2003] have published 13 advanced object-based metrics measuring aesthetic aspects of a screen—e.g., layout balance or symmetry. An example of practical application of Ngo's metrics is the QUESTIM tool designed by Zen and Vanderdonckt [2014]. Users can use it without specialized knowledge of UI design. They manually specify object boundaries according to their visual perception, and the tool calculates the values of Ngo's metrics using dimensions of the regions (Figure 1.1). The values can help them rate the overall quality of a user interface since it has been shown that aesthetics or even the first impression has an impact on usability and acceptability of the product [Tractinsky et al., 2000].

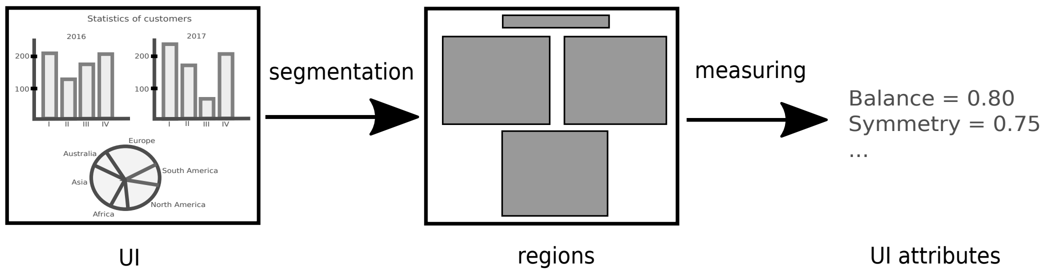


Figure 1.1: In the beginning, we have a screenshot of a user interface. We need to find a suitable segmentation method to specify regions representing visually dominant objects corresponding with the user perception. Then, we can use these regions as the inputs for object-based metrics measuring UI characteristics.

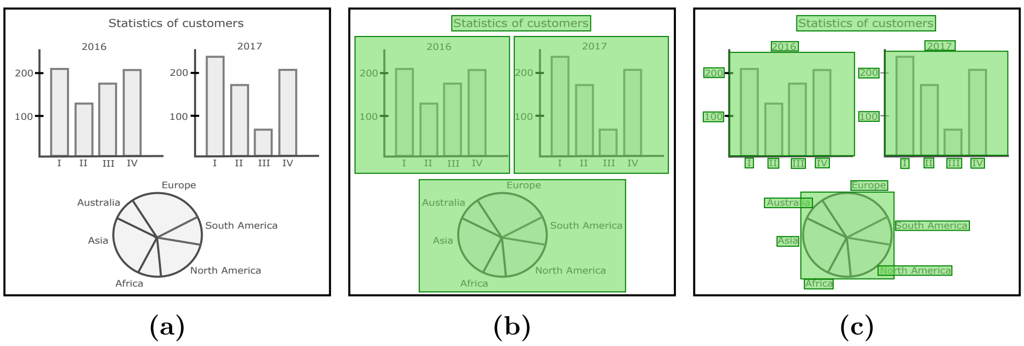


Figure 1.2: An example of the two different ways (b, c) of subjective perception of objects in a dashboard (a). The perceived objects are specified by the rectangular boundaries (regions), which are used as the inputs for object-based metrics (e.g., Balance or Symmetry).

The main weakness of the applicability of object-based metrics is the ambiguous definition of the object. The QUESTIM tool depends on the user's subjective perception of objects. Two users will most likely specify object regions in a slightly different way, which may lead to ambiguous results (Figure 1.2). There were also attempts to extract the description of objects from the structural descriptions of web-pages [Purchase et al., 2011], or images of user interfaces [Reinecke et al., 2013]. The problem with these approaches is that they do not usually consider objects with the same complexity as people perceive them (e.g., the principles of objects grouping described by Gestalt laws [Koffka, 2013]).

1.1 Goal of the Research

The goal of this research is to explore the possibility to apply the metric-based evaluation for analysis of dashboard design quality. Specifically, this research focuses on the solution of the following issues:

- Analyze the common characteristics of dashboards. Focus on the perception of objects in dashboards by users, evaluate the subjective visual perception of the users and detect the presence of Gestalt laws.
- Explore existing metrics for analysis of UI attributes and consider their application for measuring quality and usability characteristics of dashboards.
- Focus on object-based metrics of aesthetics and analyze ambiguity of measured results caused by users' subjective perception of objects.
- Create a framework for evaluation of metrics' ability to objectively distinguish well-designed dashboard samples.
- Look for a new approach which would improve the metrics' ability to distinguish well-designed dashboard samples objectively.
- Design a method for segmentation of dashboards into regions which would correspond with the average perception of the users.
- Implement a tool which would provide functionality for loading, segmentation and objective measurement and analysis of chosen dashboard characteristics.

1.2 Document Structure

The theoretical part (described in Chapter 2) of the thesis analyzes state of the art regarding data visualization, evaluation of UI, visual perception, and cognition (e.g., objects grouping and Gestalt psychology). The practical part described in Chapter 3 and 4 applies the knowledge in the field of metric-based analysis and evaluation of user interface quality. It provides own study of user perception which helps to understand the subjectivity of visual perception and object recognition and grouping in complex user interfaces like dashboards. It extends state of the art in the field of metric-based analysis, evaluation of UI and page segmentation and provides a tool for automatic analysis of dashboards and single screen UIs. The research works with static images of user interfaces, focusing strictly on the presentation aspect of UI. It does not consider the interaction of users with the analyzed user interface. Finally, Chapter 5 and 6 evaluate and summarize the thesis.

Chapter 2

State of the Art

This chapter is organized in the three sections which summarize state of the art of the thesis. **Section 2.1** introduces the dashboard visualization tool, presents existing definitions and briefly discusses its characteristics, classification, display media, design process, and design problems. **Section 2.2** provides state of the art regarding evaluation of user interfaces. It provides basic terminology and categorization of existing methods with examples. It focuses on the evaluation based on design heuristics and design guidelines based on quantitative metrics for measuring usability characteristics of user interfaces. It introduces existing pixel-based and object-based metrics and considers their application for evaluation of dashboard visual quality and points out the problem of ambiguous recognition of objects within a user interface which represent inputs for object-based metrics. **Section 2.3** focuses on the recognition of visually emphasized objects within a scene. The first part discusses the process of human visual perception of objects (e.g., Gestalt psychology, or preattentive processing, subjective perception). The second part focuses on the automatic recognition of objects by a computer. It presents existing segmentation approaches which are usually used for segmentation of scanned documents and consider their application in segmentation of dashboard screenshots.

2.1 Dashboard and Data Visualization

There are several definitions of the *dashboard* term used in information technology. Oxford Dictionary of English [Stevenson, 2010] defines it as "a graphical summary of various pieces of important information, typically used to give an overview of a business". Wexler et al. [2017] define it simply as "a visual display of data used to monitor conditions and/or facilitate understanding." Malik [2005] focuses on enterprise dashboards used for improving processes and data analysis in organizations and defines a dashboard as "a rich computer interface with charts, reports, visual indicators, and alert mechanisms that are consolidated into a dynamic and relevant information platform." Eckerson [2006] describes dashboards in the context of business intelligence as the tool called *performance dashboard*—"a multilayered application built on business intelligence and data integration infrastructure that enables organizations to measure, monitor, and manage business performance more effectively."

Looking for existing dashboards reveals us various dashboard examples, usually implemented in the form of a webpage. Wexler et al. [2017] provide a list of very different scenarios of dashboard application (e.g., monitoring of power plant operations, analysis of a patients' history of recent hospital admissions, watching the performance of Premier League players). Stephen Few [2006] examined existing dashboards, looked for common characteristics and defined a dashboard as "a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance."

The dashboards designed according to Few's design recommendations should prefer the graphical visualization of data over the textual presentation, because it can provide the overall view of the data, can emphasize important relationships between the data and can hold more data than the textual representation. They should present only the information which is important for achieving selected goals. They should fit a single screen because people can find important connections in data better if they can perceive all the data at once. They should be intuitive so that the users could perceive the information at a glance. Figure 2.1 presents an example of a dashboard which Few considers as well-designed concerning the visual design and user's requirements. As Few pointed out, most of the existing so-called dashboards break these recommendations. Business intelligence vendors use dashboards more like a marketing tool which helps them to sell the product than a tool which should actually help users to monitor and analyze data effectively.

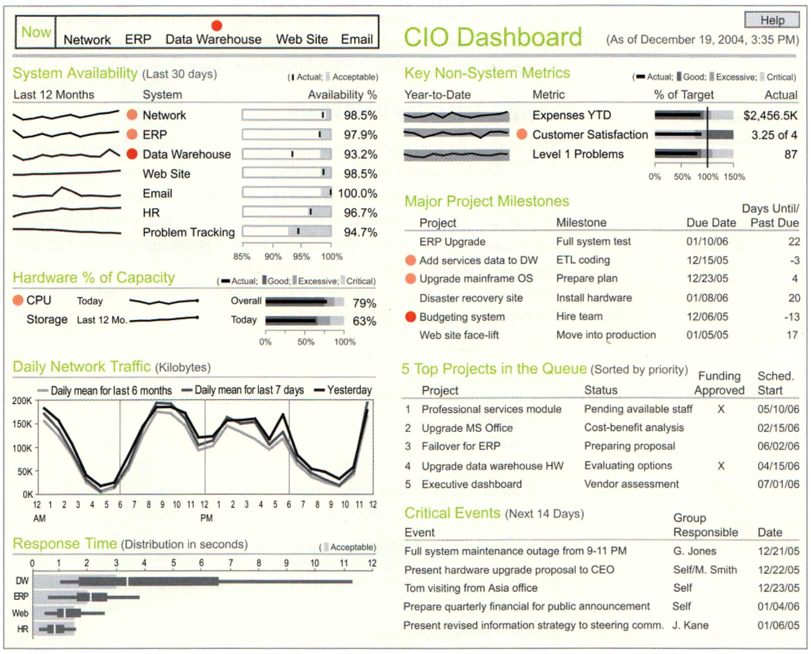


Figure 2.1: Few's CIO dashboard sample [Few, 2006]. Few reduces non-data pixels and tries to provide important contextual information, well-arranged on one screen. Red alerts inform users about real-time (the top-left table) or long-term problems and lead the users to other screens displaying the reasons of the problems.

Few [2006], Eckerson [2006] and Rasmussen et al. [2009] distinguish the three kinds of dashboards: *operational*, *analytical/tactical* and *strategic* dashboards. **Operational dashboards** provide low-level data and notify the users (usually front-line workers) about the situations which require some response. Data is simple (text values, alert icons) and it is usually updated in the real-time so the response can be performed quickly. We can find such operational dashboards for the monitoring of a stock market or traffic. **Analytical**

(tactical) dashboards provide aggregated data and help to analyze the data. They are usually used for managers and analysts. They work with a static snapshot of data and use the advantage of graphical presentation to provide a better look at the context of the data. They allow users to interact with displayed media (e.g., OLAP analysis [Codd et al., 1993; Wrembel and Koncilia, 2006]). **Strategic dashboards** provide the most summarized view containing high-level data, usually to executives who use the dashboards to analyze the progress of a strategy fulfillment. They contain static snapshots of long-term data and focus more on the performance and prediction of the future. They usually use the graphical presentation of data, but usually without the ability to provide advanced operations with the data since it is not usually required.

Dashboard screens consist of display media presenting data. There are many kinds of display media described in specialized books [Harris, 2000] or surveys [Purchase, 2014]. Few [2006] recognizes the six kinds of display media: *text*, *organizers*, *graphs*, *icons*, *drawing objects* and *images*. **Text** is used to present non-quantitative data which we cannot express numerically or to emphasize particular values. A **graph** (also called *chart* or *plot*) is the major graphical presentation used in dashboards. Few recommends to use the following graphs in dashboards: popular *bar charts*, *line charts*, *scatter plots*, and less known but effective *bullet graphs*, *sparklines*, *box plots*, and *treemaps*. He also recommends combining selected graphs to save the space and emphasize relationships between the data—e.g., bar charts with line charts. On the other hand, he does not recommend to use popular *pie charts* and *gauge charts* because of their circular shape, which might cause problems with perception of presented values. **Icons** are next important media. They are represented by symbols which usually present qualitative data which deserves attention of users (e.g., alert, up/down and on/off icons). Very useful are also **geographical maps**, which connect data with geographical location. **Drawing objects** and **images** are less typical media for dashboards. They are used in special situations.

The process of dashboard development should be, similarly to development of every other information system, based on some methodology for systems development life cycle (e.g., planning, analysis, design, implementation, testing, and maintenance [Kendall and Kendall, 2011]). The design of dashboard should not miss the following steps: analysis of dashboard purpose, definition of a dataset, selection of display media, creation of a dashboard layout, and simplification and evaluation. UI designers should determine end-users and understand their goals and needs (e.g., observing and interviewing users; creating personas, scenarios, story boards or use-case diagrams [Buley, 2013; Goodwin and Cooper, 2011; Nielsen, 1994b; Preece et al., 2015]). Management should perform a cost-benefit analysis and decide whether the dashboard is worthy of investment. Analysts should select only the data which represents the information which is important for the users (e.g., *Key Performance Indicators* (KPI) evaluating the performance of the organization [Malik, 2005; Eckerson, 2006; del-Rey-Chamorro et al., 2003]). Then, UI designers should select appropriate visual media which present the data comprehensively and arrange them on a screen. Few [2006] presents that visual media "must be the best means to display a particular type of information that is commonly found on dashboards" and "must be able to serve its purpose even when sized to fit into a small space." Finally, UI designers should evaluate design quality of the designed dashboard and adjust the UI [Tullis and Albert, 2010]. The readers interested in the implementation phase can find additional information in [Jacobs and Rudis, 2014].

As Few [2006] explains in examples, well-designed dashboards can be very helpful. On the other hand, he points out that poorly designed dashboards can lead to serious usability problems and shows most of the frequents design problems. Dashboards often exceed the boundaries of a single screen, present unimportant and inaccurate information and *non-data* pixels (e.g., decorations [Tufte, 2001]) and contain inappropriate, poorly designed or distorted display media which are arranged in inappropriate layout and present data without a clear meaning and context. The important information is often hidden. Viewers are often distracted by vivid colors or decorations. He recommends avoiding rendering charts in 3D because it makes them more difficult to read and using vivid colors in charts, which should be used only for emphasizing of important information. Charts should also be designed for color-blind people and they should not contain *chartjunk*—a term created by E. Tufte describing elements of a chart which are used only as a decoration [Tufte, 2001].

Application of Few's design heuristics and recommendations is, however, limited by the knowledge of evaluators who need to be able to understand the heuristics and apply them in a particular situation correctly. Hence, it would be useful to detect some of the design problems automatically during the design phase and decrease the time and cost of user interface evaluation.

2.2 Evaluation of User Interfaces

The problem of user experience and user interface usability has been recognized for a long time. For instance, in 1987, Ben Schneiderman published the popular book [Shneiderman, 1987] providing the list of eight golden rules of interface design. In 1994, Jakob Nielsen [1994b] described the importance of user interface usability and its impact on the acceptability of the whole system. Since then, a lot of literature describing usability and user experience has been published. We can consider usability in different contexts—e.g., software/product quality, system acceptability, or user experience.

The ISO 9241-210 standard defines usability in context of *software quality* as "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [ISO, 2010]. The ISO/IEC 25010 standard defines the 6 characteristics of usability: *appropriateness recognizability*, *learnability*, *operability*, *user error protection*, *user interface aesthetics*, *accessibility* [ISO, 2011]. Nielsen [1994b] describes the usability term in the context of *system acceptability* as "the question of how well users can use the system functionality" and characterizes it by the five attributes: *learnability*, *efficiency*, *memorability*, *satisfaction*, and *errors*.

In recent years, UI designers have mentioned the usability term frequently in the context of the *user experience* (UX) term. We can find various meanings of user experience [Forlizzi and Battarbee, 2004; McCarthy and Wright, 2007; Tullis and Albert, 2010]. Law et al. [2009] performed a survey with 275 respondents from academia and industry in order to describe the scope and characteristics of UX. The results showed that "the respondents understand the notion of user experience very differently." We can find numerous further studies dealing with the meaning of user experience [Bargas-Avila and Hornbæk, 2011; Law et al., 2014; Lallemand et al., 2015]. The international standard ISO 9241-210 defines user experience as "person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service" [ISO, 2010]. Morville [2005] created a framework to describe the seven UX facets defining product requirements: *useful*, *usable*, *desirable*, *findable*, *accessible*, *credible* and *valuable*. Roto and Rautava [2008] found the four elements: *utility*, *usability*, *social value*, and *enjoyment*. Bargas-Avila and Hornbæk [2011] analyzed existing publications describing UX from 2005 to 2009 and showed that the most common dimensions connected with UX are *emotions*, *enjoyment*, and *aesthetics*.

We can notice that many of the sources describing UX emphasize the importance of UI appearance and aesthetics. The aesthetics term is derived from the Greek *aisthanesthai*—to perceive. Merriam-Webster's dictionary [2004] explains it as "pleasurable to the senses" or "attractive." Its importance is discussed by many "non-UX" publications [Kristeller, 1951; Lavie and Tractinsky, 2004; Tractinsky, 2004]. This aspect of a UI is usually perceived very quickly before a viewer fully understands the content of the UI [Lindgaard et al., 2006]. It corresponds to various aspects—e.g., *simplicity* [Karvonen, 2000] / *complexity* [Michailidou et al., 2008]. Moshagen and Thielsch [2010] define the four facets of aesthetics: *simplicity*, *diversity*, *colorfulness*, and *craftsmanshift*. Aesthetics often plays an important role in the acceptance of a whole product. Moreover, it may improve interface usability [Kurosu and Kashimura, 1995; Tractinsky, 1997; Tractinsky et al., 2000].

UX is sometimes criticized for being vague since it is connected with fuzzy and dynamic concepts [Hassenzahl and Tractinsky, 2006; Law et al., 2009]. Characteristics like user emotions or interface aesthetics are subjective. They can change during the time as the preferences and expectations of people are changing. Either way, user experience has become an actual trend in the field of UI design and evaluation. UI designers should not overlook it.

2.2.1 Classification of Methods

The most common classification of methods presented by Ivory and Hearst [2001] or Fernandez et al. [2011] consists of the five classes: *testing*, *inquiry*, *inspection*, *analytical*, and *automation/simulation* methods: **Testing methods** are a group of methods which are based on observation of user interaction with a user interface. Examples of methods are: *think aloud protocol* [Lewis and Rieman, 1993], *A/B testing* [Siroker and Koomen, 2013], *mouse tracking* [Freeman and Ambady, 2010], *eye tracking* [Duchowski, 2007] or *performance measurements* and *log file analysis* [Andrews, 1998]. **Inquiry methods** represent a group of methods which are based on communication between an evaluator and a user. The most common methods and tools are *user feedback*, *interviews*, *surveys*, and *questionnaires* [Buley, 2013; Goodwin and Cooper, 2011; Nielsen, 1994b; Preece et al., 2015]. **Inspection methods** focus on evaluation of user interface by expert evaluators without the presence of users. The evaluators find usability problems using a predefined set of criteria or heuristics [Nielsen, 1994b,c; Hollingsed and Novick, 2007]. Examples of methods are *cognitive walkthrough* [Wharton et al., 1994], *heuristic evaluation* [1990; 1994a], or *guideline review* [Jeffries et al., 1991]. **Analytical modeling methods** provide different kinds of models which evaluators use to generate usability predictions—e.g., usability problems, the execution, or learning time. An example of a frequently used model is the GOMS model [Card et al., 2018], or KLM (*Keystroke-Level Model*) [Card et al., 1980; Kieras, 2001; Katsanos et al., 2013]. **Automation/Simulation methods** simulate user interaction using modeling languages or simulation algorithms, e.g., Petri nets or genetic algorithms [Ivory and Hearst, 2001].

Carpendale [2008] compares selected evaluation methods according to the three factors: *generalizability* (the extent to which the results of evaluation can be generalized to other users or situations), *precision* (*reliability* [Leung, 2015], the extent to which the evaluator control all aspects of evaluation—results are reliable and replicable), *realism* (the extent to which the context of evaluation is similar to the context of real usage). Improvement of one factor will decrease the level of the remaining two factors.

The Carpendale's factors correspond with *obtrusiveness* of users during an evaluation (e.g., interference between users and evaluators). The most precise methods are usually obtrusive. On the other hand, unobtrusive environment helps to generate realistic results. As pointed out by Preece et al. [2015], evaluators should combine several usability methods. Then, they can get generalizable, reliable and realistic results.

Another factor which is important for this research is the level of automation. Ivory and Hearst [2001] considers the four types of methods: *None* (a method does not support any automation of evaluation), *Capture* (a method provides the ability to capture the process of evaluation—logs of interaction with UI), *Analysis*: a method automatically detects usability problems, *Critique*: a method automatically detects usability problems and offers solutions for the problems.

The methods based on subjective feedback of users are usually without any automation support. On the other hand, the methods based on inspection or modeling of usability usually offer some level of automation. Automation provides certain advantages like decrease of time, expenses and human resources. On the contrary, the methods with a higher level of automation are usually narrow-focused. They do not consider the context of evaluation and the subjective factor of users. They detect false-positive usability problems more frequently than the methods without automation. They should be used for additional evaluation.

This research focuses on the automatic evaluation and it follows the inspection methods, particularly the evaluation based on heuristics and guidelines using quantitative metrics. The following subsections analyze selected pixel-based and object-based metrics.

2.2.2 Pixel-based Evaluation

A user interface can be implemented in various programming languages, and it can use many technologies. It might be elaborate to create a tool which would be able to work with an internal representation of a user interface. Hence, it might be useful to take a static snapshot of the screen and evaluate the UI as a raster image—e.g., measure usage of individual color values, or distribution of those values in a raster image.

Few's heuristics recommends UI designers to use subtle colors. We can evaluate this heuristic by measuring **colorfulness** of a UI snapshot. For instance, Yendrikhovskij et al. [1998] base colorfulness on the image saturation measured in the CIElab color space where the saturation is computed as the image chroma divided by the image lightness:

$$C_i = S_i + \sigma_i, \tag{2.1}$$

where S_i is the average saturation of an image i and σ_i its standard deviation. $C_i = 0$ represents achromatic image. Images with $C_i \approx 2$ can be considered highly colorful. The metric was used by Reinecke et al. [2013].

According to Few's heuristics, dashboards should contain a low number of color values, so it might be useful to analyze **the number and share of used colors** in UI. Common graphical libraries usually work with the RGB color space, which stores color values as a 24-bit number ($2^{24} = 16.77$ million distinct color values). There is a high probability, that human will not recognize all displayed color values (especially those with a low frequency of occurrence). Hence, it might be reasonable to work with posterized bitmaps. The metric was used by Purchase et al. [2012].

Inappropriate layout and distribution of graphical elements in user interface are frequent design problems [Few, 2006]. For example, we can threshold the image and measure **the distribution of the black and white pixels** or convert the image into the grayscale color space and measure **the distribution of color intensity**. Kim and Foley [1993] present a formula for measuring balance between the left and right side of a black-and-white image:

$$\text{left-right balance} = \frac{\text{total weight of less heavy side}}{\text{total weight of more heavy side}} \tag{2.2}$$

$$\text{total weight of side} = \sum 1\text{s} \cdot f(\text{distance away from center}) \tag{2.3}$$

where '1s' represents the black pixels (graphical elements) of a side in the black-and-white color space. Similarly, we could measure vertical balance or balance of an image represented in the grayscale color space (we could replace '1s' with value of normalized color intensity).

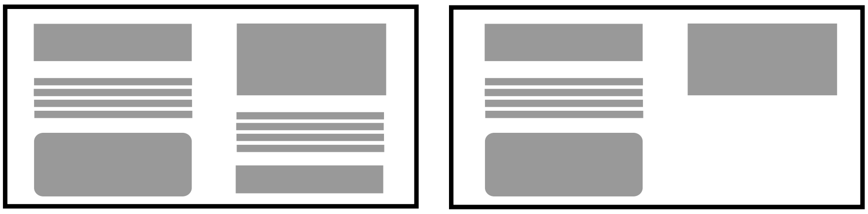


Figure 2.2: A simplified example of balanced (left) and unbalanced (right) screens.

2.2.3 Object-based Evaluation

The second approach focuses on the analysis of objects located in a user interface. Vanderdonckt and Gillo [1994] based on Foley and Van Dam [1982] recognize the two kinds of objects: *interaction* and *interactive* objects. Interaction objects (also *widgets* or *controls*) represent static (e.g., labels or separators) and dynamic (e.g., buttons, text fields) objects of a user interface. Interactive objects represent the remaining objects (e.g., drawings or pictures). Then, the rectangular boundaries of all objects (*regions*) form a layout of a user interface (Figure 2.3).

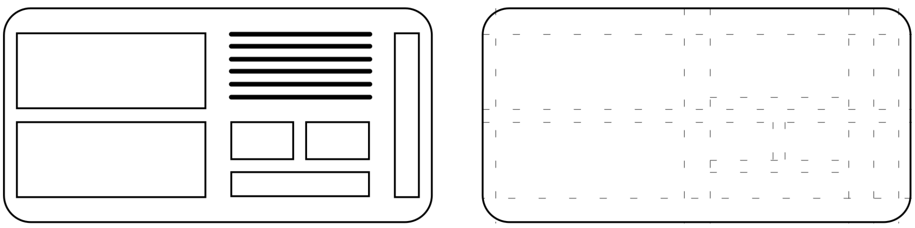


Figure 2.3: The left figure represents a simplified screen containing objects. The right figure represents the underlying layout grid. Source: Vanderdonckt and Gillo [1994].

Vanderdonckt and Gillo [1994] have published 30 advanced visual techniques for the analysis of screen layouts, divided into the five groups: *physical*, *composition*, *association* (and dissociation), *ordering* and *photographic* techniques. The techniques are described qualitatively by visual examples and descriptions. Some of them (like the physical ones) are easily convertible to an algorithm than others (like the photographic ones) which are more complex and focus on the subjective feeling of the viewer.

Quantitative measuring of object characteristics became significant with the evolution of graphical user interfaces. In the 1980s, UI designers used metrics to evaluate textual user interfaces [Smith and Mosier, 1986; Tullis, 1984]. In the 1990s, they applied metrics in the tools for the automatic design of graphical user interfaces [Ivory and Hearst, 2001]. Examples of the tools were presented by several researchers [Bodart et al., 1994; Kim and Foley, 1993; Sears, 1993, 1995; Shneiderman et al., 1995; Mahajan and Shneiderman, 1997; Parush et al., 1998]. The usual goal of the tools was to analyze simple layout properties.

In the 2000s, the rapid evolution of the Internet made UI designers focus on the evaluation of webpage user interfaces. Ivory [2001] gathered knowledge about design guidelines and heuristics until 2001 and presented the list of 157 quantitative metrics for evaluation webpage elements (e.g., analysis of the amount of text on a page, color usage, and consistency). UI designers put higher emphasis on the soft design aspects like aesthetics and the first impression of users. Ngo, Teo and Byrne [2000a, 2000b, 2001a, 2001b, 2003] attempted to describe aesthetics formally. They presented the 13 quantitative object-based metrics of aesthetics: *Balance*, *Equilibrium*, *Symmetry*, *Sequence*, *Cohesion*, *Unity*, *Proportion*, *Simplicity*, *Density*, *Regularity*, *Economy*, *Homogeneity*, *Rhythm*. The following text will refer these metrics as *Ngo's metrics*.

For instance, the *Balance* metric is defined as the "difference between total weighting of components on each side of horizontal and vertical axis." and can be calculated as:

$$\mathrm{BM} = 1 - \frac{\mid \mathrm{BM}_{\mathrm{vertical}} \mid + \mid \mathrm{BM}_{\mathrm{horizontal}} \mid}{2} \in [0,1] \tag{2.4}$$

$$\mathrm{BM}_{\mathrm{vertical}} = 1 - \frac{w_L + w_R}{\max(\mid w_L \mid, \mid w_R \mid)} \tag{2.5}$$

$$\mathrm{BM}_{\mathrm{horizontal}} = 1 - \frac{w_T + w_B}{\max(\mid w_T \mid, \mid w_B \mid)} \tag{2.6}$$

where w_j is a weighting of a side $j \in \{L, R, T, B\}$ (left, right, top, bottom) containing n_j regions:

$$w_j = \sum_{i}^{n_j} a_{ij} d_{ij} \tag{2.7}$$

The weight of a side depends on the area a_{ij} of a region and the distance d_{ij} of the region from the center of the screen. Readers can find definitions and formulas of all metrics in [Ngo et al., 2000a]. Readers can also notice the similarity of the object-based formula of Balance with the pixel-based Formula 2.2. On the contrary to the formula of the pixel-based Balance, Ngo's formulas do not consider color or shape of interface objects. They analyze a screen as a set of rectangles (regions) representing the boundaries of interface objects. The regions are described only by their dimensions (size and position). The result of every metric is a value of the [0, 1] range, which represents the rate of an aesthetic factor.

In the 2000s and 2010s, numerous researchers evaluated the applicability of Ngo's metrics to the present time, especially for website interfaces. They usually based the evaluation of the metrics on the comparison of the measured results with the reviews of users who rated user interfaces. Their results depend on a selected group of users, analyzed user interfaces and approaches to the description of interface regions. I have detected four approaches of recognition of regions.

The first approach generates its own layouts containing exact descriptions of regions. The primary purpose is to simulate specific situations used for the comparison of user perception with the results given by a metric. Examples of this approach can be found in [Altaboli and Lin, 2011; Salimun et al., 2010; Bauerly and Liu, 2008]. **The second approach** is based on the analysis of the structural description of real interfaces—e.g, analysis of document object model [Purchase et al., 2011]. **The third approach** uses raster screenshots and tries to detect regions automatically, using image processing methods. It considers the visual aspect of screen compared to the previous approaches. Examples can be found in [Zheng et al., 2009; Reinecke et al., 2013]. **The fourth approach** depends on the manual selection of regions by users. Zen and Vanderdonckt [2014] provide the QUESTIM tool, which enables the loading of a website screenshot and lets users manually specify the regions representing the input for Ngo's metrics. Other exmamples can be seen in [Zain et al., 2008; Mazumdar et al., 2015].

As described in this subsection, we can specify objects at least according to three[1] different techniques. It makes objects ambiguous as well as the results of object-based formulas. Examination of the input variables of Ngo's formulas provides us closer information about the dependency of the formulas. We can characterize Ngo's metrics by the three kinds of dependency:

- Ω_{AD}: The metrics dependent on the accuracy of areas of regions and the distribution of regions on a screen: Balance, Equilibrium, Symmetry, Sequence, Density, Rhythm and Unity. The evaluator needs to specify the parts of the screen occupied by objects accurately.

- Ω_{AR}: The metrics based on the aspect ratios of regions: Cohesion and Proportion. The evaluator needs to specify the objects' ratios of width to height accurately.

- Ω_{G}: The metrics based on the level of screen granularity (number of regions, aligned points, or different sizes): Unity, Simplicity, Regularity, Economy, and Homogeneity. The evaluator needs to divide the parts of the screen occupied by objects accurately.

The three sets Ω_{AD}, Ω_{AR}, Ω_{G} will be used in the further analyses of Ngo's metrics considering the application of the metrics for dashboards evaluation.

2.3 Recognition of Visual Components

This thesis focuses on the segmentation of dashboard screens into regions representing the visually dominant objects which can be used as inputs for object-based metrics. This section consists of two parts. The first part describes basic principles of visual perception which should be known for the segmentation of a screen. It focuses on the problem of objects recognition and grouping. The second part presents existing methods for page segmentation. It considers their applicability for the segmentation of dashboards.

[1]We can not consider the first approach which generates synthetic layouts as a technique for the description of interface regions.

2.3.1 Visual Perception of Objects

Vision is the dominant human sense. In the beginning, the visual receptors of eyes—*rods* and *cones*—detect light and send it as electrical impulses via neurons to the brain [Gibson, 1950; Ware, 2004]. The brain initially perceives the visual signal and constructs an image of the perceived view. It combines the signal detected by the three kinds of cones and interprets color values (*color subtraction*). Then, the brain detects contrasting edges and recognizes basic shapes. Human vision is much more sensitive to the differences in color and brightness than absolute brightness level [Johnson, 2010]. Color of a shape is perceived relatively to surrounding colors.

Perception of colors is subjective. Approximately 10% of population (mostly men) have the problem to distinguish certain colors [Few, 2006; Johnson, 2010]. Few [2006] recommends changing color intensity rather than color hue in a presentation to make sure that all viewers would be able to distinguish colors. For this purpose, UI designers can use alternative color models than RGB (red, green, blue)—e.g., HSB/HSL (represented by hue, saturation and brightness/lightness), or CIE L*a*b* (lightness, green-red, blue-yellow), which corresponds to human perception of colors better [Ware, 2004].

Color is one of the attributes which play a role during the initial recognition of objects and construction of the image. It is done *preattentively* according to preattentive attributes. Preattentive processing is the perceptual task of object recognition which is performed very quickly without the user's attention (in less than 200 ms according to [Healey et al., 1996]). According to Healey et al. [1996], there are 17 preattentively perceived features, which can be, according to Ware [2004], classified into the four categories: *color*, *form*, *spatial position*, and *motion*. The appropriate usage of the preattentive features can significantly decrease the time of dashboard sensemaking as shown by Few [2006] (Figure 2.4).

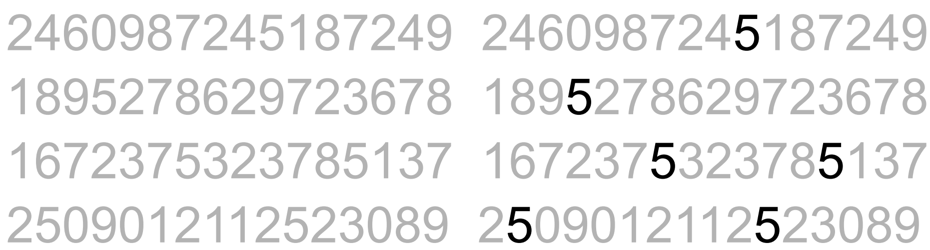


Figure 2.4: The difference between the preattentive and attentive processing. It is easier to count the number of the digit '5' in the right side because we can distinguish their different color intensity preattentively. On the contrary, we need to process the digits in the left side attentively. Based on the example presented in [Few, 2006].

After the initial recognition of objects, the brain tries to comprehend the recognized objects, organize them and add meaning to them. Baker et al. [2009] call this process *sensemaking*. He explains it as "the ability to comprehend complex information, assimilate it, create order from it, and develop a mental model of the situation as a precursor to responding to the situation." Only a fraction of what a viewer focuses on is also the object of the viewer's attention [Few, 2006]. This fact corresponds with the limited capacity of the brain's short-term memory, which stores the objects of the actual focus of attention. Few presents the size of the short-term memory between 3 and 9 items, but we can find different interpretations—e.g., 3 - 5 items according to [Johnson, 2010].

According to Baker et al. [2009], a visual representation improves sensemaking in data exploration tasks when it supports consistency with the viewer's knowledge, analogical reasoning, strong Gestalt properties, and the four basic visual perceptual approaches: association, differentiation, ordered, and quantitative perception. Viewers usually try to associate a perceived view with a previous experience or with a similar problem. Otherwise, they try to create a new interpretation of the perceived view and store it in their long-term memory. The quick recognition and comparison of objects can improve the sensemaking.

The detection of the differences and similarities between the perceived objects plays a role in object ordering and grouping. Since viewers can focus on a limited number of objects, they preattentively cluster simple graphical objects into larger visual groups and simplify the view. This fact was described by Gestalt psychology in the early 20th century [Koffka, 1922; Wertheimer, 1923; Köhler, 1925]. It presents laws describing the principles of object recognition and grouping (Figure 2.5). The problem of Gestalt laws is that they miss a mathematical model. Their quantitative description is still the aim of researchers [Jäkel et al., 2016]. This research consider the fact that a viewer will finally group simple objects into more complex visually emphasized objects.

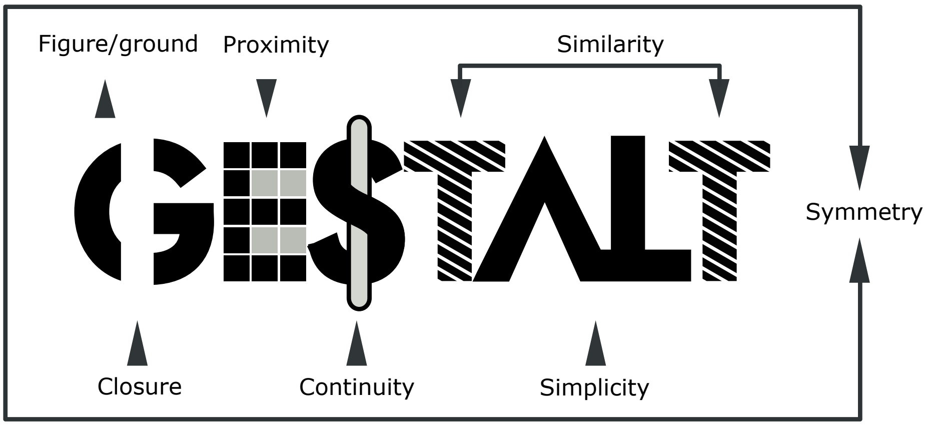


Figure 2.5: Examples of Gestalt laws. Author: Valessio S. Brito.[2]

Visually emphasized objects together with background elements (larger scale, solid surfaces, and structures) make a scene of visual representation [Henderson and Hollingworth, 1999]. Every object within a scene can be described by its visual characteristics [Baker et al., 2009]. An appropriate choice and arrangement of objects within a scene are crucial for the interpretation of data by the viewer. They can emphasize various relations between data, yet they can skew or hide other facts (examples in [Tufte, 2001]). Hence, an analysis of object characteristics within a scene can be useful during the design phase of a dashboard or user interface in general.

[2]Source: Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Gestalt.svg. The figure was translated to English.

2.3.2 Page Segmentation

Page segmentation is the important part of document processing and understanding. The goal is to divide a document page into coherent parts which can be classified and analyzed by further analyses. According to [Kise, 2014] page segmentation is "a task of extracting homogeneous components from page images." Kise [2014] considers components as text blocks or zones, text-lines, graphics, tables, and pictures. Page segmentation is usually used for digitization of printed documents or analysis of web pages. The usual reason we want to segment a page is to analyze its content, appearance, and usability.

Since we can store a page in different kinds of media (electronic or printed media), there are different approaches to process the page. Printed documents need to be scanned, so we process them as raster images. On the contrary, web pages are represented by structural description. We need to use a browser to render their *Document Object Model* (DOM) and find the nodes representing coherent parts of the page.

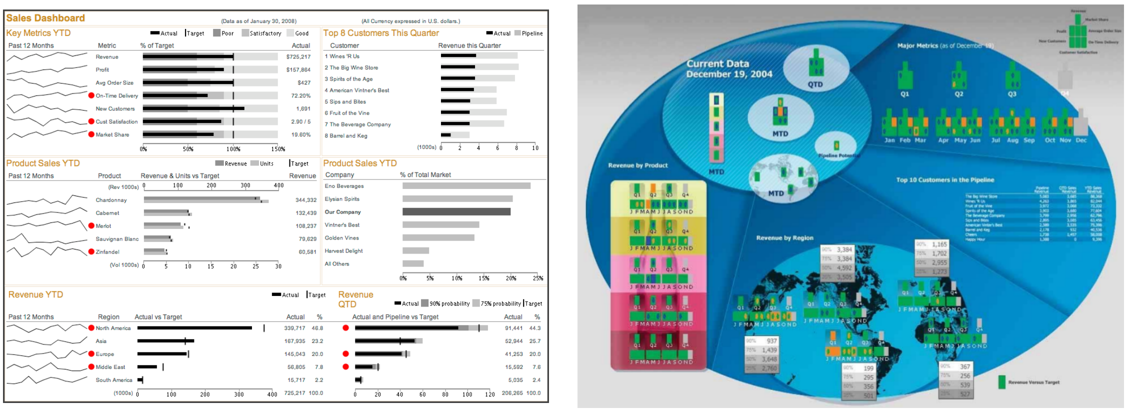
Segmentation of the pages represented by a structural description does not require to perform image processing methods (image preprocessing or OCR—*optical character recognition*). There is also no loss of quality caused by capturing of the raster image. On the other hand, a web page can contain dynamic content (JavaSript, AJAX), and some nodes can be invisible. It might be much more difficult to render the page since the result highly depends on the resolution of the screen and the browser interpreting the source code. Readers can find methods for the web page segmentation in [Burget, 2017; Feng et al., 2016].

Segmentation of the pages represented by a raster image focuses more on the way the page is presented to users than how the page is implemented. It analyzes and understands what is actually presented to users and therefore, it can predict better what is seen by the users. It, however, depends on the quality of a captured image. Applying image processing methods is usually more difficult than processing structural description. The image needs to be preprocessed and simplified.

Segmentation of raster images has been the aim of many researchers especially because of the rising need for computer processing and archiving of printed documents. Researches have developed many different methods for this purpose. Mao and Kanungo [2001]; Shafait et al. [2006] provide a methodology for performance comparison of segmentation methods. They compare the most famous methods. A comprehensive description of document image processing and recognition can be found in the book [Doermann et al., 2014]. Kise [2014] presents a thorough classification of segmentation methods according to different attributes: *page layout*, *objects of analysis*, *primitives of analysis*, and *strategy of analysis*.

Page layout can contain *non-overlapping* and *overlapping* page elements. Kise [2014] distinguishes the four layout types: *rectangular*, *Manhattan*, *non-Manhattan*, and *overlapping* layout. Analysis of the pages with the overlapping layout is significantly more difficult. There exists dashboards with overlapping elements (Figure 2.6b). The reason might be the need to fit the data into one screen or just exaggerated creativity of the designer. However, it is not common, and dashboards usually contain elements arranged in a simple non-overlapping rectangular layout or Manhattan layout.

Objects of analysis specify whether we analyze background or foreground of a page. Printed documents usually consist of a black foreground (e.g., text) and white background, which can be separated by methods based on image thresholding [Sezgin and Sankur, 2004; Russ, 2016]. On the other hand, dashboards often consist of hierarchically arranged frames, and the background is represented by multiple colors or color gradients (Figure 2.6b). We cannot use simple separation methods, e.g., thresholding. Minaee and Wang [2016] presented an example of advanced method for separation of foreground and background.



(a) (b)

Figure 2.6: An example of a dashboard with a simple layout using a reduced number of colors (left) and a highly colorful dashboard containing color gradients and overlapping widgets (right). Segmentation of the right dashboard would be more complicated compared to the left one. Source: [Few, 2006].

Primitives of analysis represent elements of the page *foreground* or *background* processed by a segmentation analysis. We can consider single pixels as primitives, but common segmentation methods usually work with larger groups—e.g., *connected components* or *projection profiles* [Kise, 2014]. This research works with the groups of same color pixels represented by their rectangular boundaries (regions). It uses heuristics to organize the regions in a tree structure representing a page layout.

A page layout consists of a hierarchy of page primitives. There are the two **strategies of the layout processing**—*top-down* and *bottom-up* strategy. The *top-down strategy* starts with a page and divides it into page elements representing leaves of the layout tree. The typical method using the top-down strategy is *Recursive XY-cut* [Nagy and Seth, 1984]. The method uses projection profiles of the page to detect gaps between the foreground pixels and splits the page into regions. Readers can find optimization of the method (e.g., [Ha et al., 1995]). On the contrary, the reversed bottom-up strategy starts with simple primitives of the page (e. g., groups of pixels) and join them into larger coherent groups. Examples are *connected components*-based methods (e.g. [Simon et al., 1997]) or *smearing*-based (also *smoothing*-based) methods [Wong et al., 1982]. Some methods combine both strategies or starts from the middle of a layout tree (*intermediate strategy*) [Kise, 2014].

There are also other factors which we need to consider—e.g., quality of a document. Since we work with user interfaces, we can assume that the samples can be captured in high quality if needed. For instance, we can convert the dashboards represented as web pages into raster images by using a headless browser (e.g., Phantom.js[3]), which can render a web page screenshot containing a specific resolution.

Finally, we also need to consider the similarity between the segmented samples. Whereas the printed documents are usually very similar, the appearance of dashboards varies in many visual aspects. There exist various dashboard templates using different layouts, widgets, colors, and styles. It complicates to design a universal segmentation algorithm. Figure 2.6 shows an example of the variability of dashboard samples.

[3]Phantom.js project's website: http://www.phantomjs.org

Chapter 3

Decomposition of Problem

The research explores the possibility to combine the knowledge described in Chapter 2—dashboards, evaluation of UIs and recognition of visual components—and create a tool which would be able to load a dashboard, convert the dashboard to an internal representation which could be used for automatic evaluation of design quality. Figure 3.1 shows the main problems of the process.

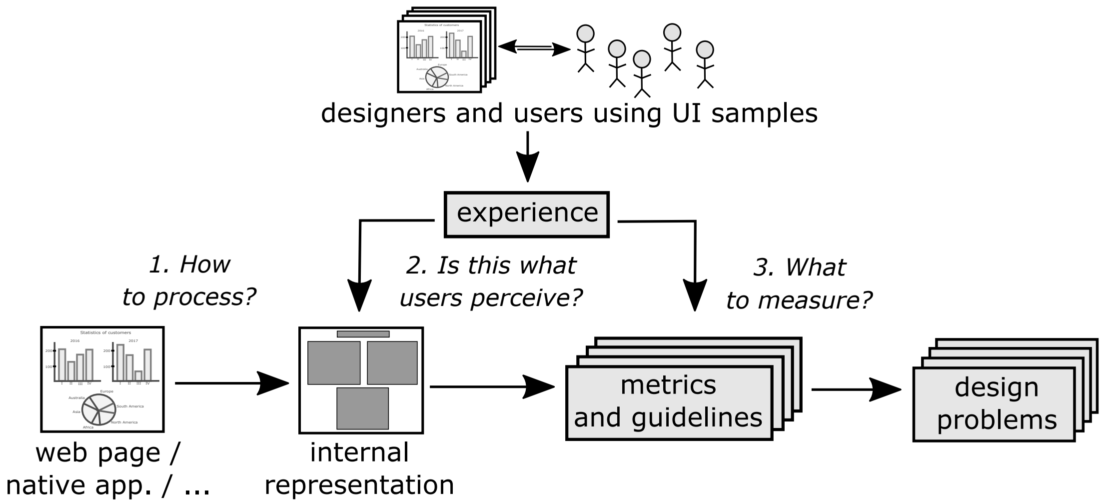


Figure 3.1: The process of UI evaluation and the main problems.

The research was split into the following tasks: specification of a model (Section 3.1), implementation of software using the model (3.2), preparation of test samples (3.3), analysis of pixel-based (4.1) and object-based (4.2) metrics, design and improvement of metrics (4.3), automatic segmentation of dashboards (4.4), comparison of metrics with user reviews (4.5).

3.1 Model

As presented in Section 2.2, we can analyze dashboards from the two perspectives: the pixel-based and object-based perspective. For this purpose, a model of dashboard was created. It defines the structure of internal representation of dashboards suitable for the evaluation of the dashboards by pixel-based and object-based metrics. The results presented in this section were published in [Hynek and Hruška, 2015].

Pixel-based representation: the first perspective represents a dashboard as a bitmap—a matrix of pixels which contain color values in a selected color space. The size of the matrix is defined by the pair: $(width, height)$, which indicates the image resolution. The RGB color space is the primary color space which is used to store dashboards. Bitmaps are then transformed into other models to reflect human perception better or reduce the number of color values. The following list presents representations of dashboard bitmaps which are used in this research: (posterized n-bit) RGB bitmap, CILEAB bitmap, HSB bitmap, (posterized n-bit) Grayscale bitmap (and histogram), Black-and-white bitmap (using a fixed or adaptive threshold). The pixel-based representations of dashboard are used as the inputs for pixel-based metrics (Sec. 4.1) and the method for segmentation of dashboards (Sec. 4.4).

Object-based representation: the second perspective considers a dashboard as a set of objects arranged on a screen which present data. The strategy of the object-based evaluation is to analyze the arrangement of objects on a screen. For this purpose, a simple theory was established. It defines the three levels of dashboard description: *model of dashboard's components*, *dashboard template*, and *dashboard sample*. The theory corresponds to the three elements of a dashboard: dashboard components (graphical elements, widgets), description of styles, and data. It is language-independent, but this research implements a simplified description of dashboard in the XML language. The description contains one root dashboard's component represented by the `<dashboard>` XML root element. This component represents a dashboard screen and contains a list of nested components representing the top-level graphical elements, as described in Listing 3.1.

Listing 3.1: The simplified description of a dashboard and regions of graphical elements without a definition of style and dataset.

```xml
<dashboard>
  <x>0</x> <!-- horizontal offset -->
  <y>0</y> <!-- vertical offset -->
  <width>1280</width> <!-- width of screen -->
  <height>1024</height> <!-- height of screen -->
  <graphicalElement>
    <type>CHART</type>
    <x>10</x>
    <y>10</y>
    <width>200</width>
    <height>100</height>
  </graphicalElement>
  <!-- further graphical elements ... -->
</dashboard>
```

3.2 Software

The three applications were used during the research: *Dashboard Analyzer*, *Webpage Download Tool*, and *Generator of Dashboard Samples*. They were developed in order to create, automatically generate, process, and evaluate dashboards (pixel-based and object-based representations). Some parts of the software were developed with the cooperation of students of Brno University of Technology. They implemented the software as practical parts of their bachelor's or master's theses supervised or consulted with the author of this thesis.

Dashboard Analyzer is a Java application which provides tools for processing and analyzing of screenshots of dashboards. It offers tools for description of dashboard components (Figure 3.2) and the APIs for implementation and evaluation of metrics measuring dashboard characteristics and algorithms for segmentation of dashboards into dashboard components. It was designed and developed by the author of this thesis, besides the two extensions provided by Adriána Jelenčíková [Jelenčíková, 2018] (webpage download tool) and Santiago Mejía [Mejía, 2018] (the bottom-up analysis of dashboards segmentation) who implemented them as part of their theses). The source code is available in Appendix B.1.

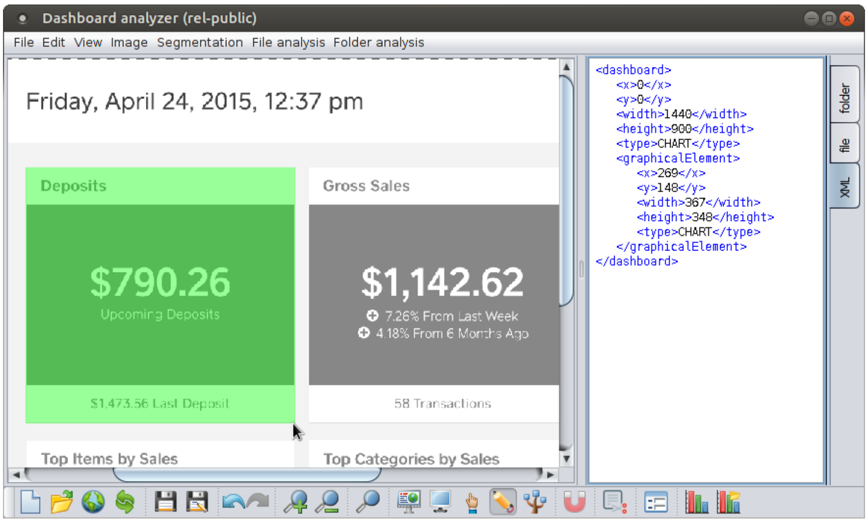


Figure 3.2: An example of the description of regions using Dashboard Analyzer. The green area represents a selection of a visual region drawn by a user. The XML source code presented on the right is re-generated with every change of the regions in the canvas. This example contains a description of the dashboard and one region.

Generator of Dashboard Samples is a web application which provides the ability to manually create or automatically generate synthetic dashboard samples represented by a bitmap and structural description. Then, the dashboards can be used for evaluation of the impact of dashboard visual characteristics on usability and quality of the dashboard. It was developed by Olena Pastushenko as part of her master's thesis [Pastushenko, 2017].

Interactive Survey Tool is a web application which helps to create interactive forms for surveying users about perceived characteristics of dashboards. The tool was developed by the author of this thesis and the source code is available in Appendix B.2.

3.3 Dataset

The evaluation of metrics required real dashboard samples in order to evaluate the real applicability of the metrics. Hence, 130 various dashboard bitmaps were gathered from the Internet. They were split into the two groups: $D_{(\text{all})} = D_{(\text{well})} \cup D_{(\text{rand})}$. $D_{(\text{well})}$ contains 9 dashboards which were designed according to the design heuristics defined by [Few, 2006]. The dashboards were considered as "well-designed". $D_{(\text{rand})}$ contains 121 randomly chosen dashboards which were collected from the Internet. No information about the usability of these dashboards was known. The dashboards were labeled as "random".

Chapter 4

Results

The model, software and dashboard samples were used to analyze the pixel-based and object metrics presented in Section 2.2, improve selected object-based metrics and design a method for segmentation dashboard, which is crucial for the conversion of dashboards into the internal representation. This chapter summarizes the process and results of the research tasks established in Chapter 3.

4.1 Analysis of Pixel-based Metrics

The goal of this research task was to analyze the visual characteristics of user interfaces which are measurable by the pixel-based metrics presented in Subsection 2.2.2. It analyzed the possibility to use the metrics for evaluation of dashboard design quality and recognition of well-designed dashboards. The results were published in [Hynek and Hruška, 2016].

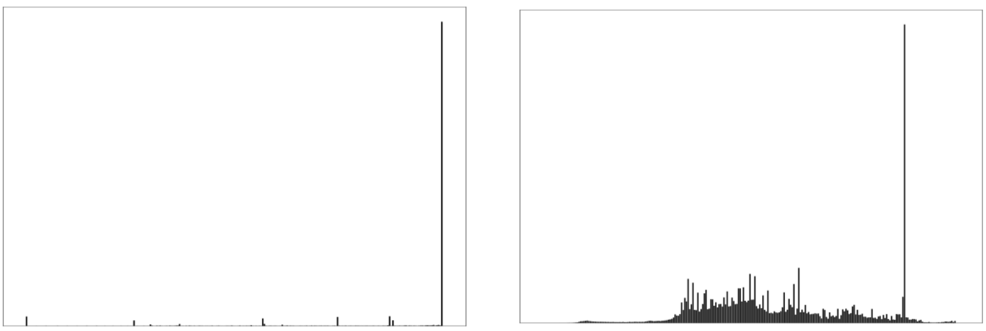
The test set was composed of the 130 various dashboard bitmaps described in Section 3.3 divided into the group $D_{(\mathrm{well})}$ of 9 well-designed and $D_{(\mathrm{rand})}$ of 121 random dashboards. Besides that, a group labeled as $D'_{(\mathrm{well})}$ was created. It contained all dashboards of $D_{(\mathrm{well})}$ resized into the 50% of the original width and height. Every dashboard was stored as a bitmap in the 32-bit RGB color space. Further transformations into other color spaces were done for the purposes of particular metrics. The metrics were implemented using Dashboard Analyzer API.

Table 4.1 presents the results for the selected metrics. It indicates that the group of well-designed dashboards is less colorful than the group of randomly chosen ones. They usually contain one or two frequently used color values, which represent background layers. Background usually occupies more than half of a well-designed dashboard. It is usually represented by some light color with a low value of saturation and high value of brightness (e.g., white). This fact can be monitored in Grayscale histograms (Figure 4.1). The results have also shown a high rate of balance and symmetry for all dashboards. The well-designed dashboards were, however, more balanced and symmetrical on average than the randomly chosen dashboards.

On the contrary, measuring the number of all colors used in RGB bitmap does not seem to be a reliable metric. The results have shown that the group of well-designed dashboards usually contains a lower number of colors than the other dashboards, but the number highly depends on the quality (resolution) of the bitmap. UI designers should keep in mind the problem regarding the ambiguous perception of color by users. Hence, it is recommended to use the posterized grayscale color space instead of the RGB color space. The metrics should be used for additional analysis providing warnings about inappropriate use of colors.

Table 4.1: The results of the analysis of chosen pixel-based metrics for the groups of dashboards: $D_{(\text{rand})}$, $D_{(\text{well})}$, and $D'_{(\text{well})}$. The values of μ represents the average value of a dashboard group and σ its standard deviation. The values of colorfulness were calculated according to Formula 2.1 designed by Yendrikhovskij et al. [1998].

Metric	$\mu_{D_{(\text{rand})}}$	$\sigma_{D_{(\text{rand})}}$	$\mu_{D_{(\text{well})}}$	$\sigma_{D_{(\text{well})}}$	$\mu_{D'_{(\text{well})}}$	$\sigma_{D'_{(\text{well})}}$
HSB						
Saturation colorfulness	0.384	0.209	0.125	0.039	0.114	0.035
CIELAB						
Saturation colorfulness	0.690	0.581	0.265	0.209	0.209	0.159
12-bit RGB						
Number of color values	677	467	250	183	374	395
Share of the 1st color	54.50%	21.27%	81.62%	8.28%	79.46%	8.18%
Share of the 1st+2nd color	66.23%	19.14%	85.97%	5.26%	84.05%	5.05%
4-bit Grayscale						
Number of color values	15.52	0.98	14.89	2.03	15.11	1.45
. . . with share > 5%	3.38	1.69	1.22	0.44	1.33	0.71
. . . with share > 10%	2.04	0.98	1.00	0.00	1.00	0.00
Share of the 1st color	57.47%	19.79%	84.81%	4.76%	82.89%	4.36%
Share of the 1st+2nd color	72.43%	18.15%	88.38%	3.55%	86.82%	3.76%
Balance	0.761	0.139	0.906	0.036	0.907	0.036
Symmetry	0.852	0.063	0.925	0.019	0.923	0.020
1-bit Black-and-white						
Share of black color	28.84%	15.80%	12.52%	3.37%	15.01%	3.44%



(a) Grayscale histogram of Fig. 2.6a

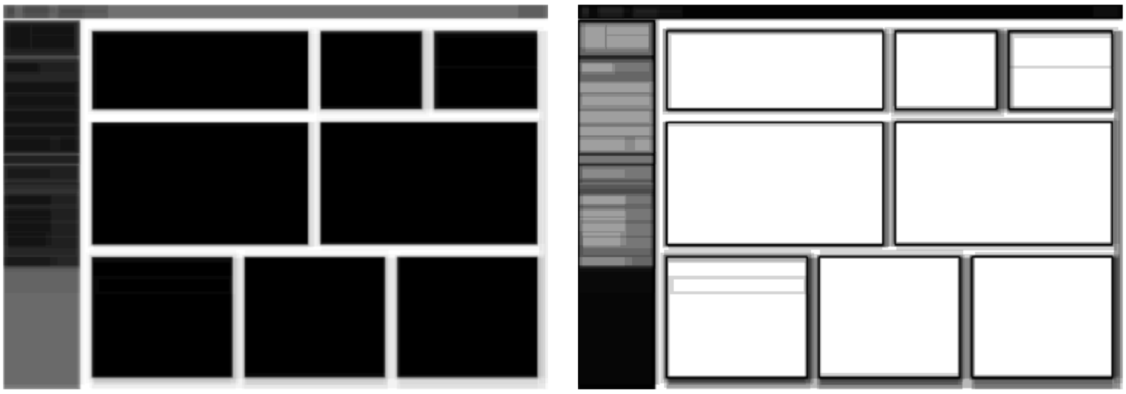
(b) Grayscale histogram of Fig. 2.6b

Figure 4.1: Histograms of the two dashboard bitmaps presented in Figure 2.6 converted to the 8-bit grayscale color space. The horizontal axis represents the values of the 8-bit grayscale color space (0-255, from black to white). The vertical axis represents the number of pixels which represent a particular color value.

4.2 Analysis of Object-based Metrics

The goal of this research task was to analyze selected visual characteristics of user interfaces which are measurable by object-based metrics. It analyzed the possibility to apply Ngo's metrics described in Subsection 2.2.3 for evaluation of dashboard design quality and recognition of well-designed dashboards. In contrast to the analysis of pixel-based metrics, the analysis of object-based metrics had to deal with the ambiguity of metrics inputs.

Firstly, the 130 dashboard bitmaps described in Section 3.3 were divided into 13 groups of 20 samples (every sample was contained by two groups). Then, the groups were uniformly distributed among 251 users who were asked to use Dashboard Analyzer in order to specify descriptions of regions representing their subjective perception of objects within a dashboard (*user description*). The user descriptions of the regions of the same dashboard were combined into one *average description* representing the probabilities $p_i \in [0, 1]$ of region occurrences for every pixel i of the dashboard. Figure 4.2a shows a visualization of such an average description in the grayscale color space.



(a) **(b)**

Figure 4.2: An example of an average description (a) and a visualization of pixel entropies (b) represented in the grayscale color space. The higher color intensity represents the higher probability (a) and higher entropy (b) of region occurrence. The pixels representing medium probabilities of region occurrence ($p_i \sim 0.5$) are represented by higher values of entropy. Such pixels usually create borders of visually dominant objects. They can also be found in management areas (toolbars, menus) on the borders of a screen.

Then, the average description was used to measure the entropy of the dashboard—a value representing the rate of user disagreement about the distribution of regions. The binary entropy of every pixel was calculated according to the following formula:

$$E_{p_i} = -(p_i \log_2 p_i + (1 - p_i) \log_2(1 - p_i)) \tag{4.1}$$

where $p_i \in \{0, 1\}$ represents the probability of region occurrence in the i-th pixel position ($i = (x, y)$) in the matrix and $E_{p_i} \in [0, 1]$. An example of visualization of entropy values can be seen in Figure 4.2b. The entropy of a dashboard was calculated as the average binary entropy of all the pixels in the dashboard. Besides that, the number of regions in dashboard and the user disagreement about this value was analyzed. For every dashboard $d \in D_{(\text{all})}$, the average number of regions μ_{ν_d} with its standard deviation σ_{ν_d} and the coefficient of variation $c_v(\mu_{\nu_d}, \sigma_{\nu_d})$ (simply c_{v_ν}) was calculated.

The average entropy of all dashboards μ_E was 0.262 ($\sigma_E = 0.109$). It means that the value $p_i = 0.955$ on average. Therefore, the average entropy can be considered as low. High entropy was detected on the borders of regions and in management areas (the black areas in Figure 4.2b). The average coefficient of variation c_{v_ν} was 0.78. It means that the standard deviations of the number of regions were relatively high compared to the average numbers of regions. Users usually agreed about the location of regions but disagreed about their quantity. They segmented the screen with different granularity, as it was suggested in Figure 1.2 in Introduction.

The second part of the research task was focused on the problem of quantifying the ambiguity of the values measured by object-based metrics. It defined a framework for processing user descriptions of regions, which are used as inputs for object-based metrics, and quantifying the ambiguity of the values measured by object-based metrics. Figure 4.3 visually explains the four steps of processing of the user descriptions of regions by a researcher:

1. At the beginning, the researcher uses a metric m to measure values for every description of regions $d_i^{(u)}$ of dashboard $d_i \in D$ provided by a users u. The values of a dashboard d_i are grouped in a set of values $V_{(d_i,m)}$. Since I worked with 130 dashboards, I created 130 sets $V_{(d_i,m)}$.

2. Then, it is appropriate to remove the values with the highest distance from the average value of $V_{(d_i,m)}$. The reason is to filter the values calculated from the most extreme descriptions of regions. I decided to remove 10% of the values of every $V_{(d_i,m)}$.

3. Every filtered set of values $V'_{(d_i,m)}$ is used to calculate the average value $\mu_{V'_{(d_i,m)}}$ (simply $\mathrm{val}_{(d_i,m)}$) with its standard deviation $\sigma_{(d_i,m)}$. Since I worked with 130 dashboards, I calculated 130 average values and standard deviations.

4. Finally, the average value $\mathrm{val}_{(d_i,m)}$ is used to calculate one average value val_m with its standard deviation λ_m. Similarly, the standard deviations $\sigma_{(d_i,m)}$ are used to calculate one average standard deviation σ_m. The procedure can be repeated for the subsets of user interfaces (e.g., well-designed and random dashboards).

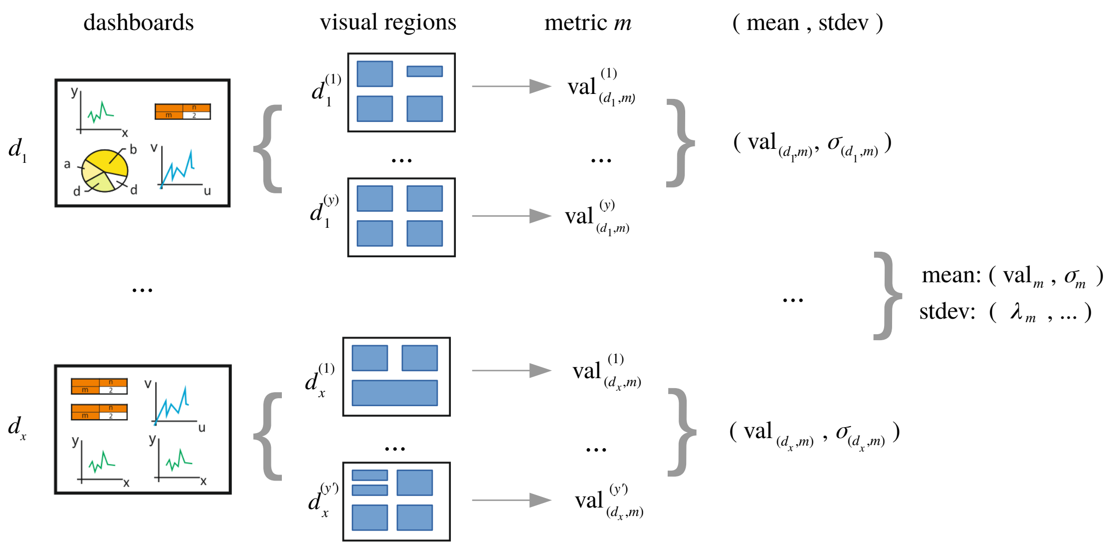


Figure 4.3: The process of measuring the values of dashboard regions ($x = 130; y \sim 39$).

Then the values σ_m and λ_m were analyzed. σ_m measures the average impact of subjective perception on the precision of a metric m. If the value of σ_m rises, there is more likely to be a greater difference between the values measured by the metric m for two independent descriptions of regions of one dashboard. I named this characteristic *metric volatility* (the opposite of *metric stability*). λ_m measures the ability of a metric m to distinguish dashboards. If the value of λ_m rises, there is more likely to be a greater difference between the values measured by the metric m for the descriptions of regions of two different dashboards. I named this characteristic *metric scalability.*

Metric stability together with metric scalability represents the characteristic which I named *metric subjectivity*:

$$\text{subjectivity}_m = \frac{\sigma_m}{\lambda_m}. \tag{4.2}$$

It measures the average impact of subjective visual perception on the precision of a metric m relative to the range of the most frequently measured values. This means that a high value of metric volatility can be compensated by a high value of metric scalability.

To rate the ability of metrics to distinguish one group of user interfaces from another (e.g., well-designed from random dashboards), the variable γ_m was established:

$$\gamma_m = \text{overlap}(\text{val}_m^{(A)}, \lambda_m^{(A)}, \text{val}_m^{(B)}, \lambda_m^{(B)}) \in [0, 1] \tag{4.3}$$

where the overlap function measures the overlapping coefficient of two normal distributions (of the groups A and B) represented by a mean val_m and a standard deviation λ_m. If the value of the overlapping coefficient λ_m rises, it will be more difficult to distinguish these two groups by the metric m.

Finally, the overall rates of the metric m are:

$$\text{objectivity}_m = \text{subjectivity}_m^{-1} = \frac{\lambda_m}{\sigma_m} \tag{4.4}$$

$$\text{decisiveness}_m = \gamma_m^{-1} \tag{4.5}$$

The more objective (stable and scalable) the metric is, the less subjectively skewed results the metric provides. The more decisive the metric is, the greater the difference between the two groups the metric can find.

The goal of the framework is to categorize and compare the metrics with each other, rather than analyze particular values of objectivity and decisiveness, which depend on the group of users and the set of analyzed samples chosen for this research. For this purpose, the following classification was established: **Class 0**—the metric m which can quantify a particular aspect of a user interface according to a specified formula; **Class 1**—the metric m of Class 0 with a *high value* of objectivity_m which is able to consider the subjectivity of visual perception to a specified extent; **Class 2**—the metric m of Class 1 with a *high value* of decisiveness_m which is able to distinguish two kinds of user interfaces to a specified extent. The definitions of Class 1 and Class 2 do not intentionally contain specifications as to what the high values of objectivity and decisiveness are because they might be different for another experiment. For this research, I set the limit of both high values to be 2.0 (λ_m will be at least 2 times higher than σ_m; γ_m will be lower than 0.5).

The last part of the research task used the framework to analyze the impact of the users' subjective perception on the ability of the 13 object-based metrics of aesthetics designed by Ngo et al. [2000a] to detect well-designed dashboards objectively. The metrics were implemented in Dashboard Analyzer. The results of analysis are shown in Figure 4.4 and 4.5.

The values of objectivity correlate with the categorization of metrics dependency described in Subsection 2.2.3. The metrics based on the analysis of screen granularity (Ω_G) have low values of objectivity, close to 1.0 (since the users segmented the screen with a different granularity). On the other hand, the values of objectivity of the metrics based on the analysis of the aspect ratios of regions (Ω_{AR}) are higher than 2.0. The remaining six metrics based on the analysis of the area and distribution of regions on a screen (Ω_{AD}) appeared to be more objective than the metrics based on the analysis of screen granularity. However, except for Rhythm, the values of their objectivity are lower than 2.0.

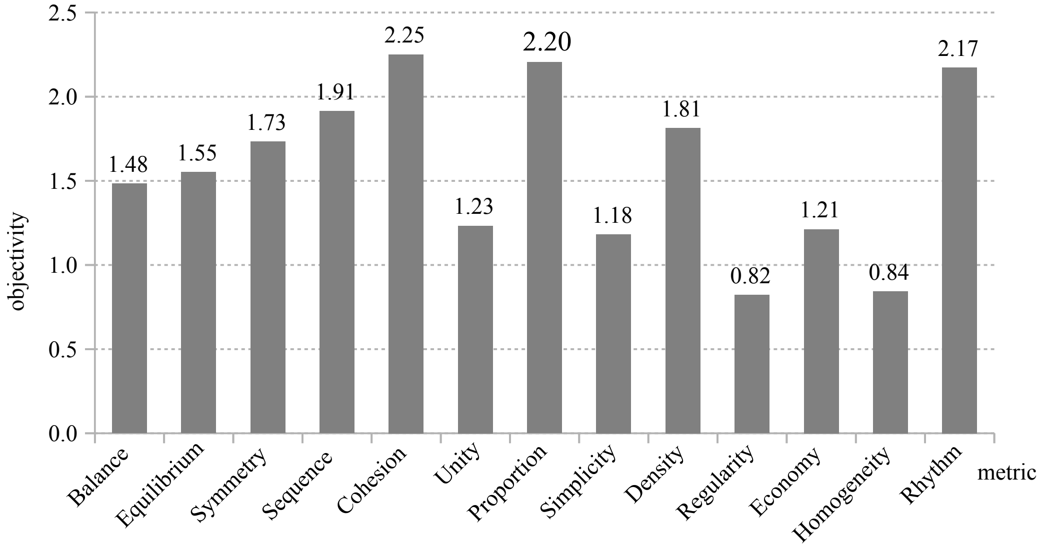


Figure 4.4: The values of metric objectivity measured for all Ngo's metrics.

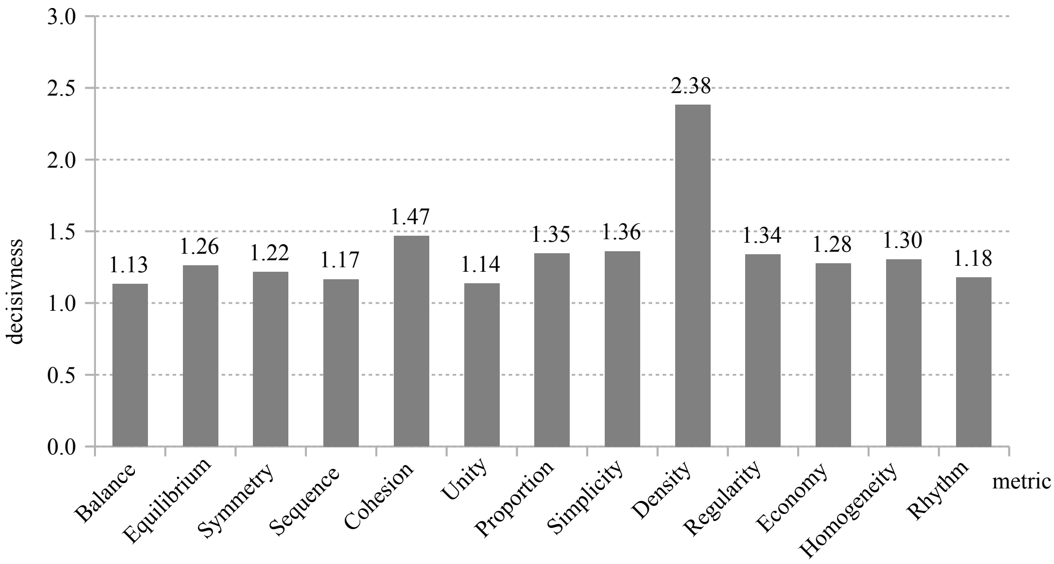


Figure 4.5: The values of metric decisiveness measured for all Ngo's metrics.

Since only three metrics were categorized as members of Class 1—Cohesion, Proportion, and Rhythm—only these metrics could become members of Class 2. However, the values of decisiveness are low, except for one metric: Density. Thus, it would be complicated to use Ngo's metrics for the detection of the well-designed samples.

4.3 Design and Improvement of Metrics

The goal of this research task was to find a solution for the problem presented in Section 4.2 regarding the inability of Ngo's metrics to distinguish the well-designed samples from the group of randomly chosen dashboards objectively. It dealt with the problem of the design of new metrics for evaluation of UI quality and improvement of existing ones. It performed a study gathering the user experience which was important for improvement of Ngo's metrics. It was performed in cooperation with Olena Pastushenko who used the study to test her Generator of Dashboard Samples (Section 3.2). The results of the study were published by Pastushenko, Hynek and Hruška [2018, 2019]. The improvement of metrics were published in [Hynek and Hruška, 2018].

First of all, the framework describing the process of improvement of metrics and design guidelines was introduced. The main idea of the framework is based on the iterative extension of a UI model representing the internal representation of the UI. In the beginning, the model represents basic information about dimensions and types of UI components. The researchers extend the model by adding new attributes representing UI characteristics which are important for the evaluation of hypotheses about the impact of the characteristics on UI quality. Then, they generate UI samples varying in the UI characteristics described in the model and let users rate the UI characteristics in order to gain a user experience. The user experience is used for further improvement of the UI model and design and improvement of metrics and design guidelines. The scheme of the framework is presented in Figure 4.6.

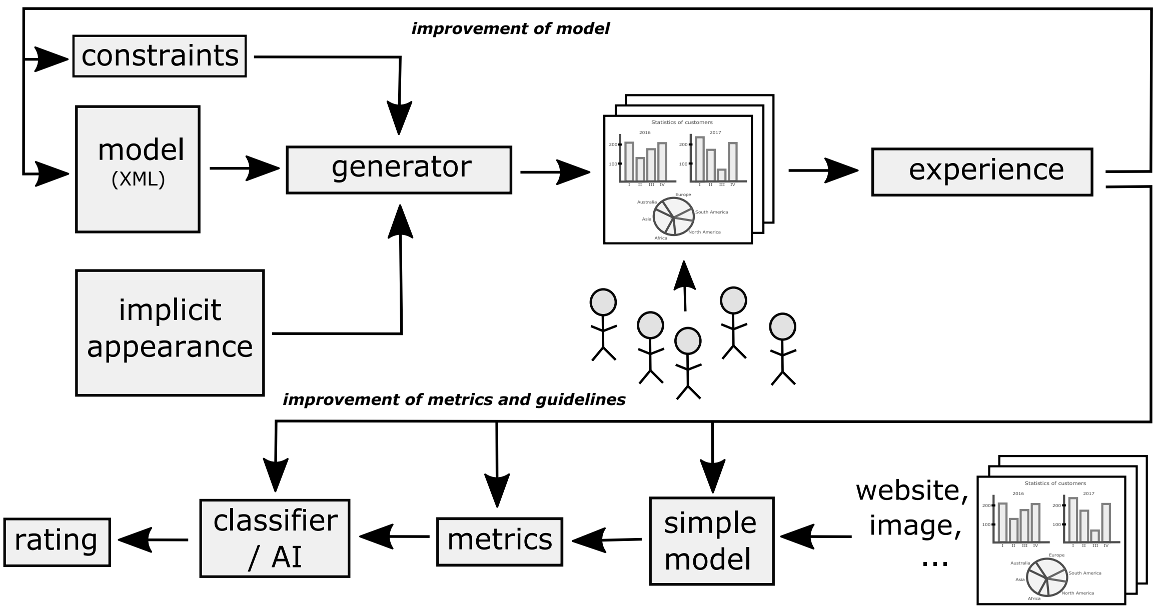


Figure 4.6: The scheme of the construction and evaluation of design guidelines. We can use the user experience for the improvement of the model (for the generation of better samples) and the improvement of the design guidelines.

The second part of the research task described a small-scale study which demonstrates usability of the framework. The study analyzed the impact of color, type of widgets, and displayed dataset on the perception of the layout balance and symmetry. It used Generator of Dashboard Samples to generate appropriate dashboard samples and Interactive Survey Tool to get the user experience. The results of the two independent user reviews (analyzing the impact on the perception of UI balance and symmetry) confirmed the impact of these factors. The users tended to perceive the charts using highly intense colors as more weighty comparing the charts using less intense colors. Similarly, the charts containing large graphical elements (e.g., bar charts) were perceived as more weighty than the charts composed of thin lines (e.g., line charts). Finally, the displayed datasets affected graphical elements of charts (e.g., size of bars), which also affected the perception of the weight of charts.

The last part of the research task used the user experience for the improvement of the object-based metrics designed by Ngo et al. [2000a]. It focused on the group of metrics based on the distribution of regions on a screen (Ω_{AD})—particularly on the Balance metric, which was rated as the least objective metric of this group. The idea of the improvement was to include objective information about the color of subjectively specified regions in the formula of Balance in order to affect weightings of the regions objectively. Hence, I modified the formula of the Balance weighting:

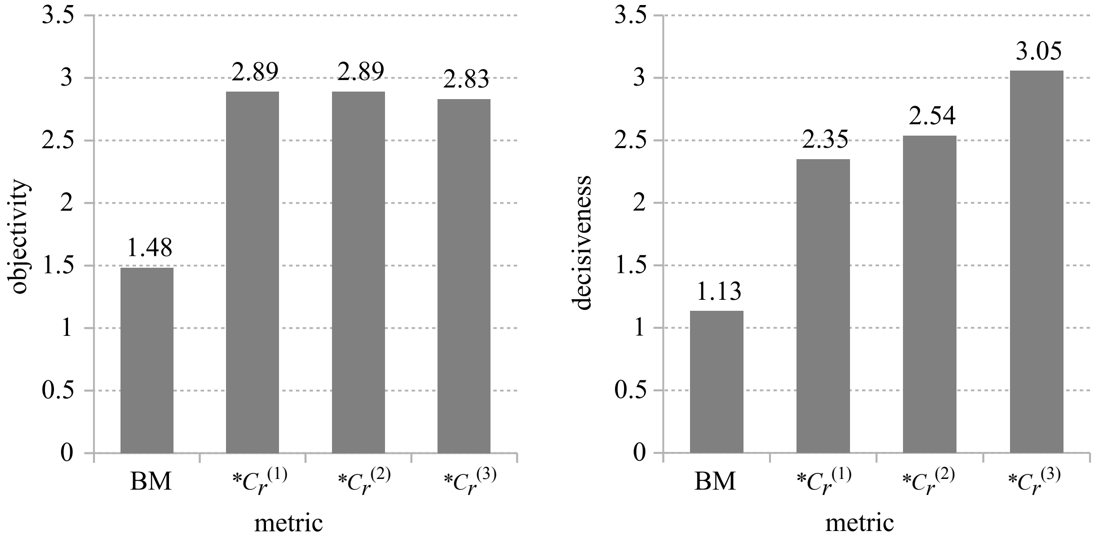
$$w_j = \sum_{i}^{n_j} a_{ij} d_{ij} C_{ij} \tag{4.6}$$

where C_{ij} is the coefficient of color of a region i in a quadrant j representing the colorfulness of the region. Since two sides of a screen are always compared to each other, there is no problem in modifying the weightings of each side by adding C_{ij} to the formula and keeping the range of the formula: $[0, 1]$. I explored several approaches to measuring the coefficient of color using different color spaces:

- $C_r^{(1)} \in [0, 1]$: the average color intensity of a region r represented in the 8-bit grayscale color space converted from the RGB color space
- $C_r^{(2)} \in [0, \inf]$: the average colorfulness of a region r inspired by [Yendrikhovskij et al., 1998; Reinecke et al., 2013] measured according to Formula 2.1
- $C_r^{(3)} \in [0, 1]$: the average value of all pixel values in a region r calculated as $1 - (b_i - b_i s_i)$ where $s_i \in \{0, 1\}$ is the saturation and $b_i \in \{0, 1\}$ is the brightness of the i-th pixel of the region in the HSB color space, based on the suggestion of [Ngo et al., 2000a]

Figure 4.7a and 4.7b present the results. We can see a significant improvement in all kinds of the coefficient of color. The values of objectivity and decisiveness are higher than 2.0, which makes Balance a member of Class 2. The best results were received for $C_r^{(3)}$, followed by the results for $C_r^{(2)}$. From a practical point of view, the easiest method improving Balance is to use the coefficient $C_r^{(1)}$ based on the color intensity, since the color intensity can be simply calculated from the RGB color space. The color intensity might also correspond better with the perception of color blind people [Few, 2006]. In addition, the infinite range of $C_r^{(2)}$ might cause problems with the modification of some metrics.

Table 4.2 presents the average values and standard deviations of Balance measured for the well-designed and randomly chosen dashboard samples. The well-designed dashboards are more balanced than the randomly chosen ones for all types of Balance, including the modified ones. This confirms the results of the pixel-based Balance presented in Section 4.1. We can see the decrease in $\mathrm{val}_{D_{(\mathrm{rand})}}$ for the modified versions of Balance. This indicates that the modified versions of Balance are stricter than the original Balance. Since the original Balance rated some dashboards as balanced, the modified versions of Balance rated the dashboards as unbalanced because of their unbalanced distribution of color on a screen.



(a) Balance objectivity **(b)** Balance decisiveness

Figure 4.7: Change of the Balance objectivity and decisiveness for the metrics using the coefficients of color: $C_r^{(1)}$, $C_r^{(2)}$, and $C_r^{(3)}$.

Table 4.2: The average values of UI balance (val) with their standard deviations (λ) for the groups of well-designed ($D_{(\mathrm{well})}$) and random dashboards ($D_{(\mathrm{rand})}$).

Metric	$\mathrm{val}_{D_{(\mathrm{well})}}$	$\lambda_{D_{(\mathrm{well})}}$	$\mathrm{val}_{D_{(\mathrm{rand})}}$	$\lambda_{D_{(\mathrm{rand})}}$
BM	0.873	0.107	0.843	0.107
BM($C_r^{(1)}$)	0.830	0.086	0.640	0.197
BM($C_r^{(2)}$)	0.819	0.072	0.643	0.182
BM($C_r^{(3)}$)	0.845	0.068	0.651	0.191

The modified version of Balance using the coefficient of color can be used for the improvement of the tools designed for metric-based evaluation of user interfaces. Since existing tools apply different approaches to detect regions, it might be appropriate to use a metric which considers possible ambiguity of the inputs. For the full automation of the evaluation, it is necessary to design a segmentation algorithm for the automatic detection of regions based on the average user perception.

4.4 Automatic Segmentation of Dashboards

The method for the segmentation of dashboards consists of several phases visualized in Figure 4.8. The results were published in [Hynek and Hruška, 2019]. The method was implemented in the Java language and integrated into Dashboard Analyzer described in Section 3.2. Readers can find the reference to the project's repository in Appendix B.1.

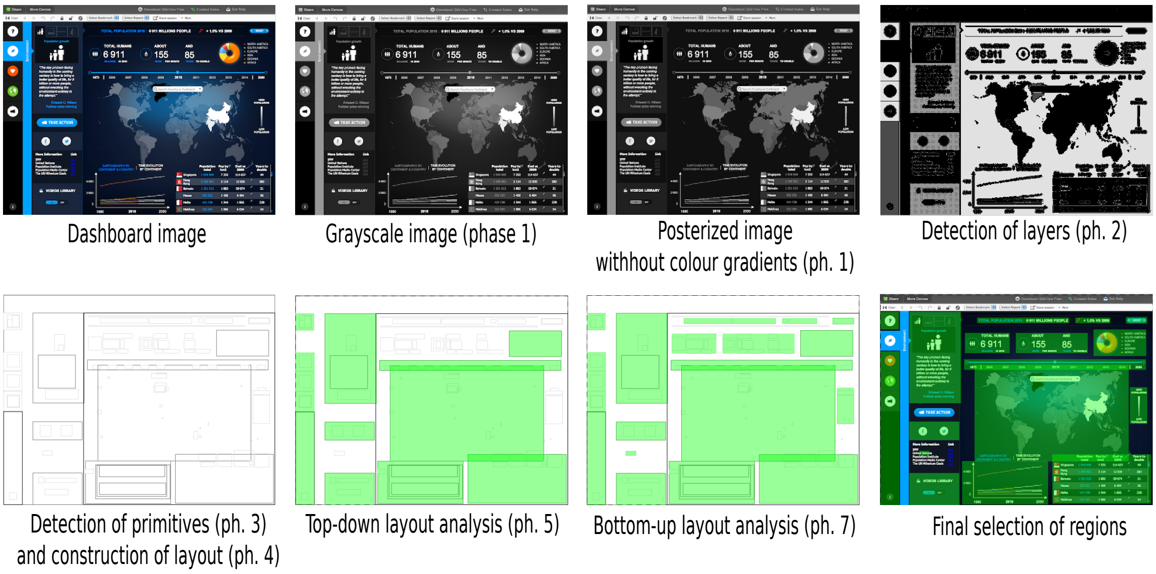


Figure 4.8: An example of the segmentation of a highly colorful dashboard containing overlapping regions. Firstly, the method preprocesses the image, reduces the number of colors and detects the color layers. Then, it constructs the layout and finds the visually dominant regions (represented by the green rectangles). Readers can notice that the method ignores some widgets, especially in the management areas. The sixth phase does not affect the regions since the dashboard does not contain any highly overlapping regions.

Phase 1—Image preprocessing: In the beginning, the method converts a dashboard bitmap into the 8-bit grayscale color space representing color intensity to reduce the number of colors to 256. Then, the method locates the areas represented by color gradients using a flood-fill-based algorithm and replace the values of all pixels of the area by the average grayscale value of the area. Finally, the method posterizes the image from the 8-bit to the [4 to 6]-bit color space.

Phase 2—Selection of colour layers: the second phase takes the preprocessed bitmap and selects (heuristically) the most frequently used colors of the bitmap. The result of the phase is a bitmap represented in *the number of dominant colors* + 1. The grayscale colors represent the layers of the bitmap which are suitable to detect page primitives and construct the page layout.

Phase 3—Detection of page primitives: the third phase detects page primitives in the bitmap. Firstly, it uses a flood-fill-based algorithm to select the areas of pixels represented by the same color (layers). Then, it converts the areas into a set of regions representing rectangular boundaries of the areas. Finally, the method filters tiny regions. It keeps the information about the layers as attributes of the regions for the top-down analysis in Phase 5.

Phase 4—Construction of layout: the fourth phase converts the set of regions into a tree structure representing the page layout. It goes through the set of regions and appends the regions into the tree, so the tree reflects the hierarchy of regions (regions are child nodes of the closest region which surrounds them). Finally, the top-level region of the tree represents the whole dashboard and the leaves represent the regions which do not surround any other regions). It is important to note that one region can be represented by more than one node in the tree (an overlapping layout).

Phase 5—Top-down layout analysis: The next phase takes the tree of regions and searches the visually dominant regions which correspond with the user perception. It analyzes the tree from the root node to the leaves and perform heuristics in order to choose large regions based on the attributes of regions from Phase 3. The method focuses on the large regions representing widget frames. It works well with the dashboards which consist of the widgets surrounded by an explicit boundary (it reflects the Gestalt law of enclosure).

Phase 6—Analysis of overlapping regions: The sixth phase detects all intersections of regions and compares the area of every intersection with the areas of the intersected regions. If the area of the intersection represents most of the area of one region (e.g., a region within another region or 2 highly overlapping regions, usually 33%), the method joins such regions into one region. Else, the method ignores the intersection.

Phase 7—Bottom-up layout analysis: The last phase focuses on the areas of a dashboard which does not contain any visually dominant region recognized in the previous phases. These areas might contain small regions which together create larger regions perceived by users with correspondence to the Gestalt law of proximity. The method takes the tree of regions representing the layout of a dashboard and analyzes it by using the bottom-up strategy. It measures the vertical and horizontal gaps between the small regions and joins the regions if the gaps are smaller than a heuristically chosen threshold. Other heuristics suitable for the bottom-up analysis were investigated in the master's thesis of Santiago Mejía supervised by the author of this thesis [Mejía, 2018].

The evaluation of the method used the set of 130 dashboard samples described in Section 3.3 and the descriptions of regions provided by 251 users described in Section 4.2. Visual comparison of the results of the segmentation with the average descriptions of regions showed that the method works well with the dashboards which contain widgets surrounded by an explicit border, which reflects the Gestalt law of enclosure well. Most of the deviations from the average perception were observed in complex dashboards which contained overlapping objects and dashboards represented in low resolution or skewed by image compression. The algorithm had occasional problems with the segmentation of management areas (e.g., toolbars and headers). It corresponds to the high disagreement of users about the description of visually dominant objects in these areas. Sometimes, the method incorrectly clustered small regions into a larger one, so the result insufficiently reflected the Gestalt law of proximity.

Quantitative comparison of $\delta_d^{(\mathrm{alg})}$ (the average difference between the values of an average description of regions and a description generated by the method) with $\delta_d^{(u)}$ (the average difference between the values of an average description of regions and a description made by a user u) showed that that the generated descriptions are at least as close to the average descriptions ($\delta_d^{(\mathrm{alg})} \leq \delta_d^{(u)}$) as 33.90% of 5,020 descriptions provided by users. Figure 4.9 shows that 119 of 130 dashboards were segmented at least as close to the average description as they were segmented at least by one user. The closer to the average description the segmentation description is, the better it reflects the perception of users.

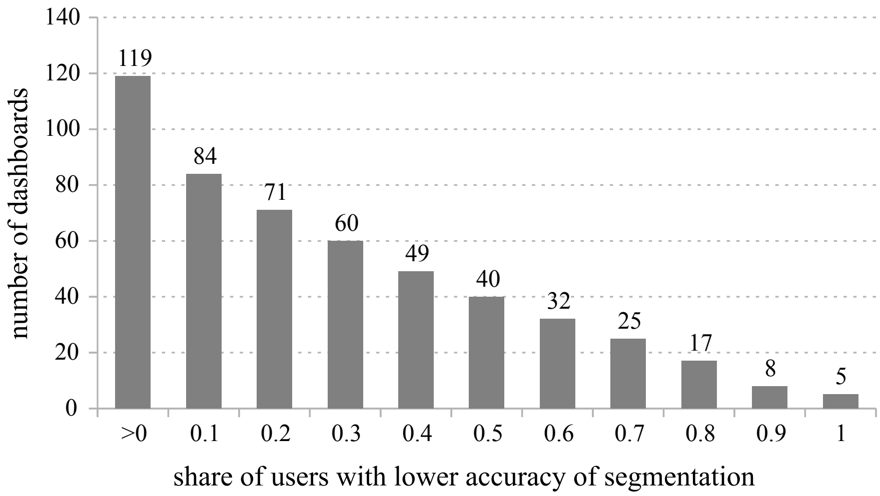


Figure 4.9: The numbers of dashboards where $\delta_d^{(\mathrm{alg})} \leq \delta_d^{(u)}$ for particular share of users. The vertical axis shows the number of dashboards. The horizontal axis represents the share of users for which $\delta_d^{(\mathrm{alg})} \leq \delta_d^{(u)}$.

Finally, Table 4.3 shows the average distance between $\mathrm{BM}_d^{(\mathrm{users})}$ (the average UI balance of users description) and $\mathrm{BM}_d^{(\mathrm{alg})}$ (the UI balance of a generated description) measured by Ngo's Balance metric and its modified versions presented in Section 4.3. The highest value was measured for the basic Balance metric: $\delta_{\mathrm{BM}}^{(\mathrm{users,alg})} = 0.100$. We can consider this value as low compared to the range of $\mathrm{BM} \in [0, 1]$. Then, we can notice the decrease of $\delta^{(\mathrm{users,alg})}$ for the modified versions of the Balance metric using the coefficients of colorfulness. Evaluators should, however, not neglect this deviation.

Table 4.3: The average distance between the $\mathrm{BM}_d^{(\mathrm{users})}$ and $\mathrm{BM}_d^{(\mathrm{alg})}$ values measured by Ngo's Balance metric and its modified versions presented in Section 4.3 for the group of all dashboards $D_{(\mathrm{all})}$. The values of $\sigma_m^{(\mathrm{users,alg})}$ represents the standard deviations of the corresponding average values.

Metric m	$\delta_m^{(\mathrm{users,alg})}$	$\sigma_m^{(\mathrm{users,alg})}$
BM	0.100	0.086
$\mathrm{BM}(C_r^{(1)})$	0.090	0.082
$\mathrm{BM}(C_r^{(2)})$	0.089	0.090
$\mathrm{BM}(C_r^{(3)})$	0.090	0.087

4.5 Comparison of Metrics with User Reviews

The goal of the last research task was to analyze user reviews of selected UI characteristics measured by the pixel-based and object-based metrics described in the previous sections. It performed two different experiments with different groups of users. The users were let to rate the selected UI characteristics. Then, the experiments analyzed ambiguity of the user reviews and correlation of the user reviews with the values measured by the metrics. The results have not been published since they should be presented in the context of the previous sections. They provide additional information and statistics regarding the perception of users and ask additional questions for further research.

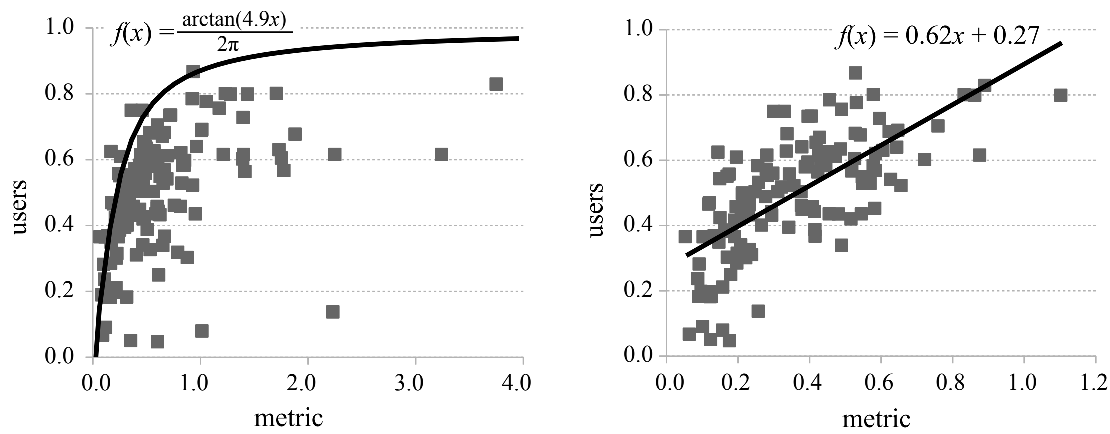
The goal of the first experiment was to collect fast data representing the perception of UI balance (*vertical* and *horizontal* using the 5-point scale: $\{-2, \ldots, 2\}$; *overall* using the 5-point scale: $\{0, \ldots, 5\}$) so the results could be used for the initial exploration of various aspects of user reviews. The experiment was performed with 36 users (35 results were valid). Every user rated one third of the 130 dashboard samples $D_{(\text{all})}$ described in Section 3.3 and corresponding description of regions $D_{(\text{BW})}$ created by thresholding from the average description of regions (Figure 4.10). They filled the ratings into a text file.

Figure 4.10: Thresholding of the average description of regions (left) according to the formula: $t = 256\frac{(p_d+(1-E_d))}{2}$, where $p_d = [0, 1]$ is the average pixel probability of regions occurrence in a dashboard d and $E_d = [0, 1]$ is the entropy of the dashboard d. The formula was designed experimentally.

The second experiment was performed as a response to the results of the first experiment. The goal was to collect more accurate, objective, and less ambiguous data than the first experiment in order to provide more accurate results regarding the correlation between the characteristics perceived by users and characteristics measured by metrics. The experiment was performed with a larger set of users ($n = 220$). The users used Interactive Survey Tool (described in Section 3.2) to rate the characteristics, which decreased the effort of the users to perform the experiment and prevented mistakes in reviews. They rated the vertical and horizontal balance similarly as the participants of the first experiment. Besides that, they analyzed the colorfulness of UI using the 5-point scale: $\{0, \ldots, 5\}$. The test set was composed of the same samples as the test set of Experiment 1: $D_{(\text{all})}$ and $D_{(\text{BW})}$. Besides that, the $D_{(\text{gray})}$ set of samples containing 130 corresponding average descriptions of regions was included. Every user rated one third of the samples.

The results have indicated the following signs of correlation:

- between the perceived colorfulness and the colorfulness measured by the two pixel-based metrics (Figure 4.11)
- between the perceived overall balance and the overall balance calculated from the values of the perceived vertical and horizontal balance
- between the values of the UI balance perceived in the real dashboard samples and the UI balance perceived in the black-and-white bitmaps representing the descriptions regions of the real dashboard samples (based on the average perception of regions by users)



(a) C_{CIELAB} **(b)** C_{HSB}

Figure 4.11: The relations between the average ratings of colorfulness and the values of colorfulness measured by the metrics.

On the other hand, the research task was unable to show the correlation between the perceived rates of balance (vertical and horizontal) and the values measured by 7 pixel-based and object-based metrics, including their combinations, trying to quantify UI balance. The users rated the visual weights of the UI sides differently. The reasons supporting their decisions should be analyzed in the future. Possible reasons might be, for instance: subjective perception of the UI characteristic, subjective quantification of the perceived rate of the UI characteristic, or subjective understanding of the UI characteristic. Sometimes, it might be difficult to provide a verbal explanation of the meaning of a metric which measures the UI characteristic. Users might consider the influence of different aspects of UI appearance. Discussion with selected people indicated that some users might have understood the balance characteristic more like the symmetry characteristic.

The gained user experience could be used either for the further improvement of the metrics or the improvement of further experiments and explanation of the characteristics to the users before the experiments. The correlation coefficients of the UI balance contradict the basic idea of the metrics proposed by Ngo et al. [2000a]; Kim and Foley [1993]; Vanderdonckt and Gillo [1994]. On the other hand, these results might not necessarily be a problem from the point of view of the automatic evaluation of UI quality. It is more important that the metrics correlate with the overall quality of UI rated by UI designers. This fact has been shown in Section 4.3.

Chapter 5

Discussion

The research explored the possibility of automatic metric-based evaluation of dashboard quality. It defined the model of internal representation of dashboards which can be processed by pixel-based or object based metrics. It provided the software dealing with the model. It analyzed the ability of chosen pixel-based and object-metrics to distinguish well-designed dashboards objectively and proposed an improvement of selected metrics. It provided the method for automatic segmentation of dashboards into the internal representation. The metrics and methods were implemented in Dashboard Analyzer, which is available in Appendix B.1.

The following sections summarize the potential applications of the results, points out limitations of the results and suggests possible future tasks which might extend the results.

5.1 Application of Results

The results can be applied in the following areas:

- Using Dashboard Analyzer (described in Section 3.2) to analyze other metrics measuring characteristics of dashboards and UIs in general. The software can be used for: further research tasks exploring the possibility to use metrics for analysis and evaluation of the UI quality; design, implementation and debugging of novel methods for segmentation of user interfaces; gathering and processing internal representations of user interfaces representing the subjective perception of UIs by users.

- Using the framework for design and improvement of metrics described in Section 4.3 (including Generator of Dashboard Samples and Interactive Survey Tool described in Section 3.2) to design novel metrics and improve existing ones.

- Using the dataset of subjective perception of regions (described in Section 4.2) for further analyses and understanding of visual perception and Gestalt laws. The descriptions of regions provide valuable information about the initial perception and interpretation of the structure of complex user interfaces by users. It can be used for observation and formalization of Gestalt laws.

- Using the method for segmentation of dashboards (described in Section 4.4) to improve existing tools for analysis of user interfaces using object-based metrics.

- Using the knowledge about the metrics (described in Chapter 4) to create tools for design and evaluation of dashboards and user interfaces in general.

5.2 Limitations

The results has the following limitations:

- The research ignores the interaction of users with dashboards. The research focused on the selected static visual aspects of dashboards. The idea is to provide additional information about the design quality of dashboards, which can be gathered automatically during the early design phase of dashboard development.
- The results of the research are based on the perception of the limited set of users. The number of users was high relatively to other researches and studies described in Section 2.2.3, but the users were similar (technical students, around 20-25 years old). The results should be verified with different types of users.
- The results of the research are based on the limited number of dashboard samples ($n = 130$). Analysis of a larger set of samples would require more time and resources. The priority of the selection of samples was finding miscellaneous dashboard samples in order to maximize the credibility of the results. The further analyses using a different set of dashboard samples should be, however, done. Also, the group of well-designed dashboard samples should contain other samples than the dashboards created according to the design guidelines described by Few [2006].
- The research was not focused on the construction of design guidelines. It analyzed the ability of the chosen metrics to distinguish the group of well-designed dashboards from the randomly chosen samples objectively. It did not specify certain limits for the rates of UI characteristics in well-designed dashboards since it is expected that these limits might vary in different situations.

5.3 Questions

The results of the research opened some questions which could be beneficial to answer in the future:

1. Is the ambiguity of the user perception of visually dominant regions generally different in specific parts of a user interface? The study of visual perception of regions (Section 4.2) suggested that some logical parts of dashboards were usually more ambiguous than the rest of the dashboard—e.g., menus or toolbars.
2. Does a high value of the ambiguity of user perception have a negative influence on the usability and quality of the user interface? It could be useful to compare the usability of user interfaces varying in the share of the area containing ambiguous parts.
3. Is it possible to improve the objectivity and decisiveness of Ngo's remaining metrics without a radical change of their characteristics? Is it possible to use all coefficients of color C_{r_1}, C_{r_2} and C_{r_3} described in Section 4.3? Dashboard Analyzer contains a suggestion for improvement of the other Ngo's metrics which are based on the area and distribution of regions on a screen. Applicability of the modified metrics should be analyzed similarly as it was done for the Balance metric and its modifications.
4. What is the optimal level of objectivity and decisiveness? The limits were established for the purposes of this research, but these levels might be different for various types of user interfaces and users. Therefore, further experiments should be done.

5. Are there any other visual characteristics (e.g., image complexity, shapes of widgets) which can be used for the improvement of metrics similarly to color?
6. How do users understand balanced and unbalanced UIs and the weight of the UI sides? Try to understand the decisions of users leading to their ratings.
7. How do the rates of the vertical and horizontal balance affect the rate of the overall UI balance? Compare the influence of the vertical and horizontal balance on the overall balance.

5.4 Suggestions for Future Work

There are various ways how to follow the results of this research. Examples are:

- Development and improvement of metrics. This research focused on the chosen pixel-based and object-based metrics measuring colorfulness and layout aspects of user interfaces. It would be useful to design metrics which would provide a more in-depth analysis of the structure and content of user interfaces (similarly to Ivory [2001]). It would, however, require an extension of the model of dashboard's internal representation and improvement of the conversion of dashboard into the internal representation.
- Improving the knowledge about user perception. Researchers should understand perception of objects within a UI closer, which might help to answer the questions: 1 and 2, or to formalize Gestalt laws, which is still the aim of researchers [Jäkel et al., 2016]. Researchers should also understand better the perception of the characteristics relating to UI quality. It might help to answer the questions: 5, 6, and 7, or to improve existing metrics, design guidelines or methods for segmentation of screen.
- Improving the credibility of the results. It can be done, for instance, by the increase in the number and variability of users, or the increase in the number and variability of UI samples.
- Development of tools for metric-based evaluation of dashboards. There are various online commercial tools which allow users to create their own dashboard. They usually do not contain any assistance which would review the design quality of the dashboard or recommends improvements of the interface. The knowledge of the metric-based evaluation of dashboards can be integrated into existing tools for dashboard design or used for the implementation of a new one as it was suggested in Section 5.1.

Chapter 6

Conclusions

Nowadays, it is not difficult to create a dashboard containing various kinds of charts by people who do not have knowledge of programming languages and databases. There are many online dashboard interactive builders which allow users to design their own dashboard quickly using a palette of predefined widgets. They, however, lack expertise in usability and overall quality of the solution. Guideline reviews based on simple quantitative metrics measuring basic design aspects offer a possibility of automatic evaluation of user interfaces.

The problem of the automatic measuring of UI characteristics is that it requires a unified UI format. The thesis proposed the language-independent model of internal representation of dashboards describing their raster and structural description suitable for pixel-based and object-based metrics. Then, this research used the model to evaluate selected pixel-based and object-based metrics.

Pixel-based metrics do not consider a user interface in the way as people perceive it. They work with a matrix of pixels while people recognize objects of the UI (simple shapes) and cluster them into logical groups which have some meaning for them (e.g., widgets, controls, or charts). They are, however, suitable for analysis of colorfulness, the most used color values, or distribution of color in a UI. Object-based metrics measure characteristics of the UI connected with the UI objects. The main problem of the object-based evaluation is, however, the vague definition of UI objects.

The research performed the study of visual perception of objects in dashboards. 251 users had provided their subjective descriptions of regions representing their perception of object boundaries in UIs. The descriptions of regions were used to analyze the ambiguity of user perception and its impact on the application of object-based metrics. For this purpose, the framework was established. It has defined the instructions on how to process the descriptions of regions and use them to calculate the values of the metric objectivity and decisiveness—the characteristics of a metric which quantify the ability of the metric to distinguish two groups of UIs objectively.

The analysis of Ngo's object-based metrics has shown that some of Ngo's metrics are not able to consider ambiguity in the perception of regions. This applies particularly to the metrics whose formula depends on the number of regions: Unity, Simplicity, Regularity, Economy, and Homogeneity. On the other hand, Cohesion and Proportion (the metrics which depend on the aspect ratios of regions) and Rhythm (a metric based on the accuracy of regions' areas and the distribution of regions) have shown higher rates of objectivity.

As the response to the results of the analysis of Ngo's object-based metrics, the framework specifying the process of design and improvement of metrics was established. It was used to improve the objectivity and decisiveness of object-based metrics. The improvement combines object-based metrics with the pixel-based approach measuring colorfulness of the interface regions. Application of the improvement to the Balance metric has shown an increase in the rate of the metric objectivity and decisiveness.

Then, the method for automatic segmentation of dashboards into regions was designed and implemented. The method uses the knowledge of the study of visual perception of objects. Particularly, it considers the Gestalt laws of enclosure and proximity, which, according to the results of the study, play a high role in the perception of regions. The method was used to segment the dashboard samples and the results were compared with the average descriptions of regions provided by the users. Most of the samples were segmented similarly to the average descriptions.

Finally, the metrics were compared with the reviews of dashboard characteristics by the two groups of 38 and 220 users. The reviews indicated high ambiguity of the values. The results showed signs of correlation between: (1) the perceived and measured colorfulness; (2) the perceived overall balance and the overall balance based on the perceived vertical and horizontal balance; (3) the UI balance perceived in the real dashboards and the black-and-white bitmaps representing the descriptions of regions of the real dashboard samples (based on the average perception). On the other hand, the research task was unable to show the correlation between the perceived and measured UI balance (vertical and horizontal). The users rated the visual weights of the UI sides differently. The reasons supporting their decisions should be analyzed in the future.

The analyzed metrics including the method for segmentation of dashboards were implemented and integrated into Dashboard Analyzer—the Java application which can load a screenshot of a user interface (from a file or URL) and analyze the screenshot using the metrics. The application provides the APIs for implementation and debugging of own metrics and methods for segmentation of UIs. The source code is available in Appendix B.1.

Future research might use the knowledge presented in this thesis to design new tools using the metric-based evaluation. For instance, it might be useful to implement an extension for a web browser which would analyze dashboard webpages. Also, the metrics could be integrated into existing dashboard builders. Besides that, a future research might focus on the improvement of existing metrics and searching for new ones. Dashboard Analyzer might be used for this purpose. The dataset of subjective perception of regions might be used for the improvement of the method for segmentation of dashboards as well as for further analyses and understanding of visual perception and Gestalt laws.

The goal of the research has been accomplished. The research has analyzed the common characteristics of dashboards and explored existing metrics for analysis of UI attributes and considered their application for measuring the quality and usability of dashboards. It has focused on the object-based metrics of aesthetics and analyzed the ambiguity of measured results caused by users' subjective perception of objects. It has created the framework for evaluation of the metrics' ability to distinguish well-designed dashboard samples objectively. It has found the new approach which improves the metrics' ability to distinguish well-designed dashboard samples objectively. It has designed the method for segmentation of dashboards into regions which correspond to the average perception of the users. It has implemented the tool which provide the functionality for loading, segmentation and objective measurement and analysis of chosen dashboard characteristics. Last but not least, the thesis has focused on the visual perception of objects in dashboards. It has evaluated the subjective visual perception of the users and detected presence of Gestalt laws. There exist many publications which focus on automatic evaluation using object-based metrics, but few of them tries to consider the subjective perception of objects by users. I see it as the major contribution of the research.

Bibliography

Altaboli, A. and Lin, Y. (2011). Investigating effects of screen layout elements on interface and screen design aesthetics. *Advances in Human-Computer Interaction*, 2011:1–10.

Andrews, J. H. (1998). Testing using log file analysis: tools, methods, and issues. In *Automated Software Engineering, 1998. Proceedings. 13th IEEE International Conference on*, pages 157–166. IEEE.

Baker, J., Jones, D., and Burkman, J. (2009). Using visual representations of data to enhance sensemaking in data exploration tasks. *Journal of the Association for Information Systems*, 10(7):533–559.

Bargas-Avila, J. A. and Hornbæk, K. (2011). Old wine in new bottles or novel challenges: a critical analysis of empirical studies of user experience. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2689–2698. ACM.

Bauerly, M. and Liu, Y. (2008). Effects of symmetry and number of compositional elements on interface and design aesthetics. *Intl. Journal of Human–Computer Interaction*, 24(3):275–287.

Bodart, F., Hennebert, A.-M., Leheureux, J.-M., and Vanderdonckt, J. (1994). Towards a dynamic strategy for computer-aided visual placement. In *Proceedings of the workshop on Advanced visual interfaces*, pages 78–87. ACM.

Buley, L. (2013). *The User Experience Team of One: A Research and Design Survival Guide*. Rosenfeld Media.

Burget, R. (2017). Information extraction from the web by matching visual presentation patterns. In *Knowledge Graphs and Language Technology, LNCS*, volume 10579, pages 10–26.

Card, S. K., Moran, T. P., and Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410.

Card, S. K., Moran, T. P., and Newell, A. (2018). *The psychology of human-computer interaction*. CRC Press.

Carpendale, S. (2008). Evaluating information visualizations. In *Information visualization, LNCS*, volume 4950, pages 19–45. Springer.

Charfi, S., Trabelsi, A., Ezzedine, H., and Kolski, C. (2014). Widgets dedicated to user interface evaluation. *International Journal of Human-Computer Interaction*, 30(5):408–421.

Codd, E. F., Codd, S. B., and Salley, C. T. (1993). *Providing OLAP (On-line Analytical Processing) to User-analysts: An IT Mandate.* Codd & Associates.

del-Rey-Chamorro, F. M., Roy, R., van Wegen, B., and Steele, A. (2003). A framework to create key performance indicators for knowledge management solutions. *Journal of Knowledge Management*, 7(2):46–62.

Doermann, D., Tombre, K., et al. (2014). *Handbook of Document Image Processing and Recognition.* Springer.

Duchowski, A. (2007). *Eye Tracking Methodology: Theory and Practice.* Springer.

Eckerson, W. W. (2006). *Performance Dashboards: Measuring, Monitoring, and Managing Your Business.* Wiley.

Feng, H., Zhang, W., Wu, H., and Wang, C.-J. (2016). Web page segmentation and its application for web information crawling. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 598–605. IEEE.

Fernandez, A., Insfran, E., and Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8):789–817.

Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data.* O'Reilly.

Foley, J. D. and Van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics.* Addison-Wesley.

Forlizzi, J. and Battarbee, K. (2004). Understanding experience in interactive systems. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 261–268. ACM.

Freeman, J. B. and Ambady, N. (2010). Mousetracker: Software for studying real-time mental processing using a computer mouse-tracking method. *Behavior Research Methods*, 42(1):226–241.

Gibson, J. J. (1950). *The perception of the visual world.* Houghton Mifflin.

Goodwin, K. and Cooper, A. (2011). *Designing for the Digital Age: How to Create Human-Centered Products and Services.* Wiley.

Ha, J., Haralick, R. M., and Phillips, I. T. (1995). Recursive X-Y cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pages 952–955. IEEE.

Harris, R. L. (2000). *Information Graphics: A Comprehensive Illustrated Reference.* Oxford University Press.

Hassenzahl, M. and Tractinsky, N. (2006). User experience-a research agenda. *Behaviour & information technology*, 25(2):91–97.

Healey, C. G., Booth, K. S., and Enns, J. T. (1996). High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(2):107–135.

Henderson, J. M. and Hollingworth, A. (1999). High-level scene perception. *Annual Review of Psychology*, 50(1):243–271.

Hollingsed, T. and Novick, D. G. (2007). Usability inspection methods after 15 years of research and practice. In *Proceedings of the 25th annual ACM international conference on Design of communication (SIGDOC '07)*, pages 249–255. ACM.

Hynek, J. and Hruška, T. (2015). Automatic evaluation of information dashboard usability. In *Proceedings of Second International Conference on Advances in Information Processing and Communication Technology - IPCT 2015*, pages 129–133. Institute of Research Engineers and Doctors.

Hynek, J. and Hruška, T. (2016). Pixel-based analysis of information dashboard attributes. In *ADBIS 2016: New Trends in Databases and Information Systems, CCIS*, pages 29–36. Springer.

Hynek, J. and Hruška, T. (2018). Application of object-based metrics for recognition of well-designed dashboards. *International Journal of Human–Computer Interaction*, pages 1–13, in press, doi: 10.1080/10447318.2018.1518004.

Hynek, J. and Hruška, T. (2019). Segmentation of dashboard screen images: Preparation of inputs for object-based metrics of ui quality. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: IVAPP*, pages 199–207. INSTICC, SciTePress.

International Organization for Standardization (ISO) (2010). *ISO 9241-210:2010 Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems*. ISO.

International Organization for Standardization (ISO) (2011). *ISO/IEC 25010:2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models*. ISO.

Ivory, M. Y. (2001). An empirical foundation for automated web interface evaluation. Dissertation thesis (Supervisor: Hearst, M.), University of California, Berkeley.

Ivory, M. Y. and Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516.

Jacobs, J. and Rudis, B. (2014). *Data-driven security: analysis, visualization and dashboards*. Wiley.

Jäkel, F., Singh, M., Wichmann, F. A., and Herzog, M. H. (2016). An overview of quantitative approaches in gestalt perception. *Vision Research*, 126:3–8.

Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. (1991). User interface evaluation in the real world: a comparison of four techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*, pages 119–124. ACM.

Jelenčíková, A. (2018). Analysis and processing of information dashboards. Bachelor's thesis (Supervisor: Hruška T.), Brno University of Technology.

Johnson, J. (2010). *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. Elsevier.

Karvonen, K. (2000). The beauty of simplicity. In *Proceedings on the 2000 conference on Universal Usability (CUU '00)*, pages 85–90. ACM.

Katsanos, C., Karousos, N., Tselios, N., Xenos, M., and Avouris, N. (2013). KLM Form Analyzer: Automated evaluation of web form filling tasks using human performance models. In *Human-Computer Interaction – INTERACT 2013, LNCS*, volume 8118, pages 530–537. Springer.

Kendall, K. E. and Kendall, J. E. (2011). *Systems analysis and design*. Prentice Hall.

Kieras, D. (2001). Using the keystroke-level model to estimate execution times. Technical report, University of Michigan.

Kim, W. C. and Foley, J. D. (1993). Providing high-level control and expert assistance in the user interface presentation design. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, pages 430–437. ACM.

Kise, K. (2014). Page segmentation techniques in document analysis. In *Handbook of Document Image Processing and Recognition*, chapter 5, pages 135–175. Springer.

Koffka, K. (1922). Perception: an introduction to the Gestalt-theorie. *Psychological Bulletin*, 19(10):531–585.

Koffka, K. (2013). *Principles of Gestalt Psychology*. The International Library of Psychology. Taylor & Francis.

Kristeller, P. O. (1951). The modern system of the arts: A study in the history of aesthetics part I. *Journal of the History of Ideas*, 12(4):496–527.

Kurosu, M. and Kashimura, K. (1995). Apparent usability vs. inherent usability: experimental analysis on the determinants of the apparent usability. In *Conference Companion on Human Factors in Computing Systems (CHI '95)*, pages 292–293. ACM.

Köhler, W. (1925). An aspect of Gestalt psychology. *The Pedagogical Seminary and Journal of Genetic Psychology*, 32(4):691–723.

Lallemand, C., Gronier, G., and Koenig, V. (2015). User experience: A concept without consensus? Exploring practitioners' perspectives through an international survey. *Computers in Human Behavior*, 43:35–48.

Lavie, T. and Tractinsky, N. (2004). Assessing dimensions of perceived visual aesthetics of web sites. *International Journal of Human-Computer Studies*, 60(3):269–298.

Law, E. L.-C., Roto, V., Hassenzahl, M., Vermeeren, A. P., and Kort, J. (2009). Understanding, scoping and defining user experience: a survey approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, pages 719–728. ACM.

Law, E. L.-C., van Schaik, P., and Roto, V. (2014). Attitudes towards user experience (UX) measurement. *International Journal of Human-Computer Studies*, 72(6):526–541.

Leung, L. (2015). Validity, reliability, and generalizability in qualitative research. *Journal of family medicine and primary care*, 4(3):324–327.

Lewis, C. and Rieman, J. (1993). *Task-centered User Interface Design: A Practical Introduction*. University of Colorado, Boulder, Department of Computer Science.

Lindgaard, G., Fernandes, G., Dudek, C., and Brown, J. (2006). Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & Information Technology*, 25(2):115–126.

Mahajan, R. and Shneiderman, B. (1997). Visual and textual consistency checking tools for graphical user interfaces. *IEEE Transactions on Software Engineering*, 23(11):722–735.

Malik, S. (2005). *Enterprise dashboards: design and best practices for IT*. Wiley.

Mao, S. and Kanungo, T. (2001). Empirical performance evaluation methodology and its application to page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):242–256.

Mazumdar, S., Petrelli, D., Elbedweihy, K., Lanfranchi, V., and Ciravegna, F. (2015). Affective graphs: The visual appeal of linked data. *Semantic Web*, 6(3):277–312.

McCarthy, J. and Wright, P. (2007). *Technology as Experience*. MIT Press.

Mejía, S. (2018). Analysis of dashboard attributes based on automatic decomposition of screen. Master's thesis (Supervisor: Hynek J.), Brno University of Technology.

Merriam-Webster Inc. (2004). *Merriam-Webster's Collegiate Dictionary*. Merriam-Webster.

Michailidou, E., Harper, S., and Bechhofer, S. (2008). Visual complexity and aesthetic perception of web pages. In *Proceedings of the 26th annual ACM international conference on Design of communication (SIGDOC '08)*, pages 215–224. ACM.

Minaee, S. and Wang, Y. (2016). Screen content image segmentation using sparse decomposition and total variation minimization. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3882–3886. IEEE.

Morville, P. (2005). Experience design unplugged. In *ACM SIGGRAPH 2005 Web program*, page 10. ACM.

Moshagen, M. and Thielsch, M. T. (2010). Facets of visual aesthetics. *International Journal of Human-Computer Studies*, 68(10):689–709.

Nagy, G. and Seth, S. (1984). Hierarchical representation of optically scanned documents. In *Proceedings of the 7th International Conference on Pattern Recognition*, pages 347–349. IEEE Computer Society Press.

Ngo, D., Teo, L., and Byrne, J. (2000a). Formalising guidelines for the design of screen layouts. *Displays*, 21(1):3–15.

Ngo, D. C. L. (2001). Measuring the aesthetic elements of screen designs. *Displays*, 22(3):73–78.

Ngo, D. C. L. and Byrne, J. G. (2001). Application of an aesthetic evaluation model to data entry screens. *Computers in Human Behavior*, 17(2):149–185.

Ngo, D. C. L., Samsudin, A., and Abdullah, R. (2000b). Aesthetic measures for assessing graphic screens. *Journal of Information Science and Engineering*, 16(1):97–116.

Ngo, D. C. L., Teo, L. S., and Byrne, J. G. (2003). Modelling interface aesthetics. *Information Sciences*, 152:25–46.

Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing (CHI '94)*, pages 152–158. ACM.

Nielsen, J. (1994b). *Usability Engineering*. Academic Press.

Nielsen, J. (1994c). Usability inspection methods. In *Conference Companion on Human Factors in Computing Systems (CHI '94)*, pages 413–414. ACM.

Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing (CHI '90)*, pages 249–256. ACM.

Parush, A., Nadir, R., and Shtub, A. (1998). Evaluating the layout of graphical user interface screens: Validation of a numerical computerized model. *International Journal of Human-Computer Interaction*, 10(4):343–360.

Pastushenko, O. (2017). Web application for information dashboard design. Master's thesis (Supervisor: Hynek J.), Brno University of Technology.

Pastushenko, O., Hynek, J., and Hruška, T. (2018). Generation of test samples for construction of dashboard design guidelines: Impact of color on layout balance. In *World Conference on Information Systems and Technologies*, pages 980–990. Springer.

Pastushenko, O., Hynek, J., and Hruška, T. (2019). Evaluation of user interface design metrics using generator of realistic-looking dashboard samples. *Expert Systems*, pages 1–19, in press, doi: 10.1111/exsy.12434.

Preece, J., Rogers, Y., and Sharp, H. (2015). *Interaction Design: Beyond Human-Computer Interaction*. Wiley.

Purchase, H. C. (2014). Twelve years of diagrams research. *Journal of Visual Languages & Computing*, 25(2):57–75.

Purchase, H. C., Freeman, E., and Hamer, J. (2012). An exploration of visual complexity. In *Diagrams 2012: Diagrammatic Representation and Inference, LNCS*, volume 7352, pages 200–213.

Purchase, H. C., Hamer, J., Jamieson, A., and Ryan, O. (2011). Investigating objective measures of web page aesthetics and usability. In *Proceedings of the Twelfth Australasian User Interface Conference - Volume 117 (AUIC '11)*, pages 19–28. Australian Computer Society.

Rasmussen, N. H., Bansal, M., and Chen, C. Y. (2009). *Business Dashboards: A Visual Catalog for Design and Deployment*. Wiley.

Reinecke, K., Yeh, T., Miratrix, L., Mardiko, R., Zhao, Y., Liu, J., and Gajos, K. Z. (2013). Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*, pages 2049–2058. ACM.

Roto, V. and Rautava, M. (2008). User experience elements and brand promise. In *International Engagability & Design Conference, in conjunction with NordiCHI'08 conference.*

Russ, J. C. (2016). *The Image Processing Handbook.* CRC Press.

Salimun, C., Purchase, H. C., Simmons, D. R., and Brewster, S. (2010). Preference ranking of screen layout principles. In *Proceedings of the 24th BCS Interaction Specialist Group Conference (BCS'10)*, pages 81–87. British Computer Society.

Sears, A. (1993). Layout appropriateness: A metric for evaluating user interface widget layout. *IEEE Transactions on Software Engineering*, 19(7):707–719.

Sears, A. (1995). AIDE: A step toward metric-based interface development tools. In *Proceedings of the 8th annual ACM symposium on User interface and software technology (UIST '95)*, pages 101–110. ACM.

Sezgin, M. and Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–166.

Shafait, F., Keysers, D., and Breuel, T. M. (2006). Performance comparison of six algorithms for page segmentation. In *Document Analysis Systems VII, LNCS*, volume 3872, pages 368–379.

Shneiderman, B. (1987). *Designing the user interface: strategies for effective human-computer interaction.* Addison-Wesley.

Shneiderman, B., Chimera, R., Jog, N., Stimart, R., and White, D. (1995). Evaluating spatial and textual style of displays. Technical report, University of Maryland.

Simon, A., Pret, J.-C., and Johnson, A. P. (1997). A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):273–277.

Siroker, D. and Koomen, P. (2013). *A/B Testing: The Most Powerful Way to Turn Clicks Into Customers.* Wiley.

Smith, S. L. and Mosier, J. N. (1986). Guidelines for designing user interface software. Technical report, Defense Technical Information Center.

Stevenson, A. (2010). *Oxford Dictionary of English.* Oxford Dictionary of English. Oxford University Press.

Tractinsky, N. (1997). Aesthetics and apparent usability: empirically assessing cultural and methodological issues. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '97).* ACM Press.

Tractinsky, N. (2004). Toward the study of aesthetics in information technology. In *Proceedings of the 25th International Conference on Information Systems (ICIS 2004)*, pages 771–780. AIS Electronic Library.

Tractinsky, N., Katz, A. S., and Ikar, D. (2000). What is beautiful is usable. *Interacting with Computers*, 13(2):127–145.

Tufte, E. (2001). *The Visual Display of Quantitative Information*. Graphics Press.

Tullis, T. and Albert, W. (2010). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Elsevier.

Tullis, T. S. (1984). Predicting the usability of alphanumeric displays. Dissertation thesis, Rice University.

Vanderdonckt, J. and Gillo, X. (1994). Visual techniques for traditional and multimedia layouts. In *Proceedings of the workshop on Advanced visual interfaces (AVI '94)*, pages 95–104. ACM.

Ware, C. (2004). *Information Visualization: Perception for Design*. Elsevier.

Wertheimer, M. (1923). A brief introduction to gestalt, identifying key theories and principles. *Psychologische Forschung*, 4(1):301–350.

Wexler, S., Shaffer, J., and Cotgreave, A. (2017). *The Big Book of Dashboards: Visualizing Your Data Using Real-world Business Scenarios*. Wiley.

Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In *Usability Inspection Methods*, chapter 5, pages 105–140. Wiley.

Wong, K. Y., Casey, R. G., and Wahl, F. M. (1982). Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656.

Wrembel, R. and Koncilia, C. (2006). *Data Warehouses and OLAP: Concepts, Architectures, and Solutions*. IRM Press.

Yendrikhovskij, S., Blommaert, F. J., and de Ridder, H. (1998). Optimizing color reproduction of natural images. In *The 6th Color and Imaging Conference: Color Science, Systems, and Applications*, pages 140–145. Society for Imaging Science and Technology.

Zain, J. M., Tey, M., and Goh, Y. (2008). Probing a self-developed aesthetics measurement application (SDA) in measuring aesthetics of mandarin learning web page interfaces. *International Journal of Computer Science and Network Security*, 8(1):31–40.

Zen, M. and Vanderdonckt, J. (2014). Towards an evaluation of graphical user interfaces aesthetics based on metrics. In *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12. IEEE.

Zheng, X. S., Chakraborty, I., Lin, J. J.-W., and Rauschenberger, R. (2009). Correlating low-level image statistics with users-rapid aesthetic and affective judgments of web pages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, pages 1–10. ACM.

Appendix A

Attachments

All attached files are available in the online repository:
https://github.com/jirka/dash.thesis

A.1 Dataset

The repository contains the datasets which were used in the research tasks: dashboard samples, descriptions of regions, and user reviews of the selected UI characteristic. Identifiers of the users were anonymized.

A.2 Workspace

The repository contains the workspace for the analyses which were performed in the particular research tasks using the dataset of Appendix A.1:

- the scripts for preparation of the workspace used by Dashboard Analyzer for analyses of metrics and methods for segmentation of dashboards
- the scripts for analysis of the users reviews

A.3 Results

The repository contains the `.odt` files summarizing the results of the analyses (Appendix A.2) corresponding to the particular research tasks described in Chapter 4.

Appendix B

Software

B.1 Dashboard Analyzer

The source code, description, and license terms are available in the online repository:
https://github.com/jirka/dash

The source code contains implementation of:

- the pixel-based metrics described in Section 4.1
- the Ngo's object-based metrics described in Section 4.2
- the modified versions of the Balance metrics described in Section 4.3
- the method for the segmentation of dashboards described in Section 4.4

The software allows users to generate the results and values presented in this research using the dataset presented in Appendix A.

B.2 Interactive Survey Tool

The source code, description, and license terms are available in the online repository:
https://github.com/jirka/survey-tool

Curriculum Vitae

Personal Data

Name: Jiří Hynek
Birth year: 1989
E-mail: ihynek@fit.vut.cz
Website: http://www.fit.vut.cz/~ihynek
ORCID: 0000-0002-7292-6094

Education

2004 – 2008: Gymnázium Žamberk
secondary general education school

2008 – 2011: Brno University of Technology: Faculty of Information Technology
Bachelor's Degree Programme: Information Technology
thesis: *Profiling Perspective in Eclipse*; (in cooperation with Lissom)

2011 – 2013: Brno University of Technology: Master's degree
Master's Degree Programme: Management and Information Technologies
thesis: *CodAL Language Editor in Eclipse Framework*; (Lissom)

2013 – present: Brno University of Technology: Faculty of Information Technology
Doctoral Degree Programme: Computer Science and Engineering

Certificates

2013: IPMA: Certified Project Management Associate (Level D)

Work experience

practical experience with development of UI:

2013 – 2014: Lissom: part-time developer in the IDE team
2014 – 2017: Codasip: part-time developer in the IDE team
2017 – present: Codasip: consultant, agreements to complete job

practical experience with research, publishing, supervising and teaching:

2013 – present: Brno University of Technology: technical stuff, researcher
project member: IT4Innovations excellence in science – LQ1602
publishing: 4 international proceedings and 2 IF journals
supervising: 15 bachelor's and 5 master's theses
teaching: C language basics, information systems/web technologies