Perceptual Grouping of Digital Sketches

David Lindlbauer

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

INTERACTIVE MEDIA

in Hagenberg

im Oktober 2012

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, October 8, 2012

David Lindlbauer

Contents

Acknowledgments

Kurzfassung

Digitales Zeichnen bietet viele Vorteile gegenüber dem Arbeiten auf Papier, da die Inhalte auch nach der Erstellung verändert werden können. Um Elemente einer Zeichnung modifizieren zu können, müssen diese vorher selektiert werden. Dies kann vor allem bei der Selektion mehrerer Elemente mühsam sein. Vor allem auf großen digitalen Displays ist das Problem der Ermüdung bekannt. Obwohl Menschen ohne Aufwand visuell zusammengehörige Elemente und Strukturen sehen können, hilft diese Fähigkeit nicht bei Selektionen in digitalen Zeichnungen. Um die Selektion dieser Strukturen einfacher und müheloser zu machen wurde *Suggero* entwickelt. Suggero analysiert digitale Zeichnungen und findet wahrnehmungsbasierte Elementgruppen anhand verschiedener Eigenschaften wie räumliche Nähe oder Gleichheit von Form und Farbe. Integriert als Selektionswerkzeug in einer digitalen Zeichenapplikation gibt Suggero Vorschläge von visuell zusammengehörigen Gruppen. Zwei Benutzerstudien wurden durchgeführt. Die erste Studie gibt Einblicke in die Erwartungshaltung von Benutzern an diese Art von Selektionswerkzeugen. In der zweiten Studie wurde Suggero mit einem regulären Selektionswerkzeug verglichen. Die Resultate der Studie legen nahe, dass Suggero sowohl die Anzahl an nötigen Benutzeraktionen, als auch die nötigen Bewegungen zur Durchführung der Selektion verringert.

Abstract

Digital sketches offer great advantages over pen-and-paper sketches due to the possibility to modify content after its creation. These modifications often require many tedious selections before a particular editing tool can be applied. These interactions can be especially fatiguing on a large interactive wall. Humans have the ability to easily see perceptually related element in sketches, but cannot take advantage of it when performing selections. To address this issue, a method is proposed to facilitate the selection process, called *Suggero*. This method first identifies groups of perceptually related drawing elements during sketch interactions. These "perceptual groups" are then used to suggest possible selection extensions in response to a user's initial manual selection. Two user studies were conducted. First, a background study investigated users' expectations of such a selection assistance tool. Then, an empirical study compared the effectiveness of Suggero with an existing manual selection technique. Study results revealed that selections required fewer pen interactions and less pen movement when Suggero was available, suggesting that Suggero helps minimize fatigue during digital sketching.

Chapter 1

Introduction

Digital sketching environments offer great possibilities for creating and modifying content. Allowing users to modify their sketches during and after creation can be tremendously useful compared to paper-based sketching. However, performing such modifications can still be time consuming and cumbersome with existing digital sketching tools. Many sketching applications require selection of drawing elements before a desired modification can be made to those elements. Depending on the desired target elements, performing the selection can be tedious and time consuming. For elements overlapped or covered by other elements or in visually complex sketches, users often have no other possibility than to select the elements in small subsets or even one by one. Since digital sketches can be created on many devices like desktop computers, mobile phones, tables or large wall displays, the difficulties and problems performing such selections vary. The problem is exacerbated on a large wall display, such as an interactive whiteboard, as arm fatigue can occur if many tedious selection actions are required. Being able to perform fast and effortless selections is important for many applications which allow users to create and modify digital sketches.

The choice which elements to modify is often guided by the way a sketch and its elements are seen. We perceive the elements of a drawing as visually connected, based on different perceptual features like proximity or similarity [10, 31, 33, 41–44]. In sketches, these *perceptual groups* are often the target of modifications such as moving, rotating or recoloring. Humans are tremendously skilled at visually identifying these related drawing elements (see Figure 1.1). This work finds these relations in sketches automatically and assist users when performing selections.

1.1   Human Perception of Groups and Objects

The human perception of groups and objects has been a subject of extensive research for a long time. People are able to perceive groups and objects

Figure 1.1: A sketch contains many visual groups, based on features like proximity, shape similarity or color similarity. Selections of some of these groups can be tedious (like e.g. selection of all trees) and require much effort.

instantaneously and basically effortless. Although this perceptual process is natural for humans and its results are often found as being obvious, there is no exact knowledge on how human group perception works [32].

One of the most well known theories on human perception and perceptual grouping is the Gestalt theory proposed by the members of the Berlin School of Experimental Psychology around 1920 [10, 44]. Several *Gestalt laws of grouping* were introduced based on different factors like proximity and similarity. Examples of the Gestalt laws of grouping would be that spatially close objects or objects with similar colors are more likely to be perceived as belonging together. These laws, although lacking any kind of mathematical model, were used as the foundation of many works [9, 18, 37, 40]. One reason for this could be that the Gestalt laws, often referred to as Gestalt rules or principles (as in [32]), seem to be modelling the obvious. By not including a model for proximity or similarity or any ranking of the importance of these factors, there are many different ways of interpreting, modelling and implementing the Gestalt principles of grouping [32].

Another, more recent, theory on human group perception is the *Feature Integration Theory*, proposed by Treisman and Gelade around 1980 [41–43]. They stated that the process of visual grouping is a so-called preattentive process, meaning that groupings and objects structures are perceived and processed in the first few hundred milliseconds of a visual sensation. Several important *features* were found that seem to be important for visual grouping, for example color or shape.

The concept of *visual features* is very important, since these features are

the visual properties that connect the elements of sketches to form the perceptual groups. Not only the concepts found by Treisman can be referred to as features, but also the different visual concepts like proximity and similarity from Gestalt theory.

Previous research on supporting users by automatically finding perceptual groups, often referred to as *perceptual grouping* or *perceptual organization*, used these and other theories as a foundation. By including the knowledge from perceptual psychology and psychophysics, researchers tried to model human perception or find groupings that are perceived as visually salient and sound.

1.2   Perceptual Grouping

To support users by finding perceptual groups, previous research has investigated different techniques. As one of the first research in this area, Lowe [25] used Gestalt theoretical findings for visual recognition of objects. Although the information found by this system was nor directly presented to users, the work showed interesting possibilities and applications in terms of usage of features like proximity or symmetry. Igarashi et al. [18], Saund et al. [37] and Shipman et al. [38] tried to infer perceptual groups and structures from drawings and diagrams. The found structures were used to clean up drawings or presented to users for further interactions. In recent years, the interest in finding perceptual groups and use them for user interaction has increased. Works by Thórisson [40] and Rome [36] show the use of feature extraction and analysis with the goal of finding visually salient groups. Dehmeshki and Stuerzlinger [9] supported users by extracting visual features from user selection gestures and use them to find perceptual groups. Grossman et al. [13] provided suggestions of perceptual groups based in users manual selections.

All works focused either on perceptual groups, selection gestures, or interaction. This research explores the ability of a selection assistance tool to leverage similar perceptual grouping principles in order to identify and suggest possible selection options during digital sketching.

1.3   Suggero

In this thesis, we present *Suggero*, a tool to assist users performing selections of perceptual groups in digital sketches. The work is based on insights from Gestalt Theory [10, 44] and the Feature Integration Theory [41–43] to identify perceptual groups in real-time during digital sketching. These groups are then used to suggest possible selection extensions when a user begins a manual selection (see Figure 1.2). Suggero is designed to analyze the contents of hand-drawn digital sketches with the goal to provide users with suggestions of perceptual groups. A three phase approach is used to achieve this
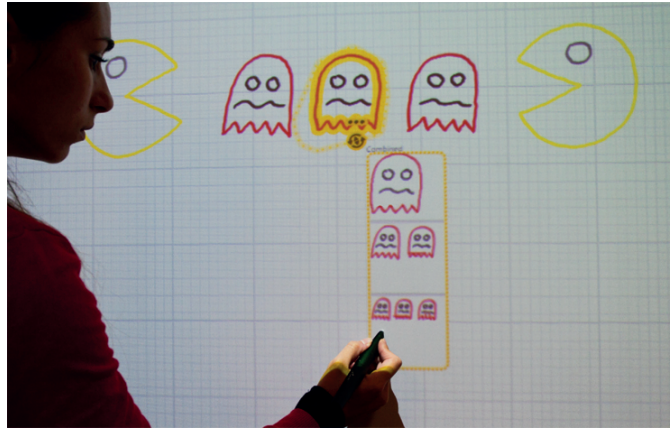
Figure 1.2: Suggero in action on an interactive whiteboard.

goal, with the phases being *Pre-processing*, *Feature Extraction* and *Dynamic Grouping*. The result of the Dynamic Grouping is a collection of perceptual groups, which is presented as *suggestions* based on users' manual selections.

1.3.1 Pre-processing

In the *Pre-Processing*, the input of the digital sketching environment is prepared for the next two phases. Suggero works with different digital input devices (e.g. mouse, digital pen or touch). The resulting input are collections of 2D coordinates (the input strokes). Due to this fact, no pre-processing in terms of stroke detection or stroke extraction is needed. Suggero is triggered every time a new element is added to the sketch or an existing element is modified (e.g. moved, rotated, removed). For the later Dynamic Grouping, properties like the bounding circle (2D bounding sphere) and Fourier descriptors of each stroke are computed and persisted. This brings great increases in performance, because the pre-processed features don't have to be recomputed in the grouping phase.

1.3.2 Feature Extraction

In the *Feature Extraction*, the pairwise relations between all elements are computed. The features *proximity*, *shape similarity*, *color similarity*, *stroke thickness similarity*, and *parallelism* are analyzed. For each feature, values indicating the pairwise element relations, the *affinity values*, are calculated. With these values it is possible to express the relation between pairs of elements without relying on absolute decisions, for example the decision if two elements are close together or similar. The values serve as input for the next phase, the Dynamic Grouping.

1.3.3   Dynamic Grouping

In the *Dynamic Grouping*, the output from the Feature Extraction is analyzed. Suggero uses a similarity based grouping approach, influenced and implemented with the *Hierarchical agglomerative clustering* algorithm [27]. The drawing elements are grouped pairwise, based on their relation. Suggero searches elements with the closest relations and assigns them to groups. All groups found by this algorithm are ranked based on their quality. The quality is determined by calculating the average affinity value of all elements in a group. The result of the Dynamic Grouping is a ranked collection of groups, which are the final result of the three phases, the *perceptual groups*.

1.3.4   Suggestions

Since sketches consist of many different perceptual groups, it is neither possible nor practical to present all of the automatically found groups at once. Also, automatically selecting groups for users would result in more harm than help, since Suggero is not able to read users' minds. Thus, it is not possible to know exactly what an intended selection may be. Also, as Findlater and McGrenere [11] reported, users prefer to maintain some control over the system. They tested completely manual menus, automatically generated and adaptable menus. Their result showed that users preferred the adaptable menus, which were automatically generated but could be modified. These findings can also be applied to the interaction with perceptual groups. Instead of simply selecting the highest ranked group, Suggero presents a few suggestions in form of a menu next to the manually selected stroke. In the context of this thesis, two different menus were implemented, a vertical list and a marking menu, both presenting users with the three highest ranked suggestions. This way, Suggero assists users without them loosing control over their selection. Suggero is integrated in a sketching application running on a large vertical display, an interactive whiteboard.

1.4   Contributions

The contribution of this thesis include detailed descriptions of all three phases of Suggero and all algorithms used. In order to evaluate the proposed system, two user studies were conducted. To explore the user expectations of a perceptual grouping system, we conducted a preliminary background study. Then, to examine the impact of using perceptual grouping for selection assistance, we conducted a comparative study, comparing our proposed selection technique Suggero with an existing sketch-based selection method, called Harpoon [24]. In order to compare Suggero with Harpoon, a method was developed that allows fair and unbiased evaluation. The results showed that Suggero decreases the amount of interactions needed to perform se-

lections, which is important to avoid effects of fatigue, especially on large display.

1.5 Outline

The thesis is structured as follows. In Chapter 2, background information on perceptual grouping is provided. Related work regarding perceptual psychology and the two influential theories of perception used in this thesis—the Gestalt laws and the Feature Integration Theory—is provided. Additionally, past research is presented, which used the knowledge from perceptual research to support users by finding perceptual groups.

The proposed system Suggero and its concepts are presented in Chapter 3. The workflow, as well as the choice of perceptual features and their applications are explained. Finally in this chapter, the interactions with the system and its provided perceptual groupings are shown.

In Chapter 4, an in depth look into the implementation of the system is presented. The algorithms used in the system to extract the important features from a digital sketch are explained. Afterwards, the dynamic grouping process is shown and how the extracted features are used to find visually salient groups from the input elements.

The preliminary background study is presented in Chapter 5, the comparative user study in Chapter 6. Both chapters included details on the experimental design, as well as results and discussion. Chapter 7 gives a conclusion of the thesis and an outlook on future work.

Chapter 2

Related Work

This chapter presents an overview of influential and important perceptual psychology research used in Suggero. An overview of past research on perceptual grouping is provided.

2.1 Perceptual Psychology

Suggero largely depends on two theories from perceptual psychology, namely Gestalt theory and the Feature Integration Theory. Both are presented in the next sections.

2.1.1 Gestalt Laws

The Gestalt theory (or Gestalt laws) was developed around 1920 as part of Gestalt psychology by Wertheimer, Koffka and Köhler and other members of the Berlin School of Experimental Psychology [10, 44]. The theory addressed human perception of visual input, including the perception of objects—or wholes—and the perception of visual structures. After exploring the human vision, several laws and rules were formulated to explain when humans perceive objects as structures or groups and when objects stand for themselves. Some of the laws require prior knowledge of the drawing or are based on past experiences of viewers. Others do not require any context and thus are more applicable to context agnostic analysis like perceptual grouping. Wertheimer and others explored several laws and over the years, other researchers have proposed new laws to fit human perception, for example Palmer's principle of common region [31] or Palmer and Rock's principle of connectedness [33]. The gestalt laws include theories on object perception based on factors like *proximity, similarity, closure, good continuity, common fate, symmetry, figure and ground, past experience*, and *Prägnanz*.

Gestalt laws are sometimes referred to as Gestalt principles or factors, because they do not provide any mathematical model [32]. There is no measure

of e.g. the law of common fate. Even the law of proximity is hard to measure, because there is no fixed threshold or value when two objects are perceived as being close together of far apart. It may seem obvious that close or similar objects are perceived as being part of the same structure. This could be one reason, why the laws are still accepted and often mentioned in literature. They, to a certain extent, describe the for humans obvious nature of vision, which is hard to describe in exact mathematical models but is clearly existing for humans [32].

Over the years, other theories on perceptual organization were developed and explored. They deal with different factors of vision, like properties of objects in perceptual groups or when the process of perceptual grouping takes place. One well known theory is the so called *Feature Integration Theory of Attention*, developed by Treisman and Gelade [41–43]. Their theory of feature maps, that are created in the preattentive stage of vision, deals with both, the time when and the process how groups and structures are created in the human vision system.

2.1.2   Feature Integration Theory

Treisman and Gelade [43] developed the feature-integration theory, based on previous work and later extended by Treisman [41, 42]. The theory states, that humans process several features simultaneously in a preattentive stage of vision. In [42], Treisman stated that

> "If grouping is an early, preattentive process, it should be mediated only by the discrimination of simple, separable features."

Several experiments showed that participants could easily see and remember structures that consisted of elements sharing same features like color or orientation. Treisman [41] noted that

> "...boundaries are salient between elements that differ in simple properties such as color, brightness and line orientation but not between elements that differ in how their properties are combined or arranged."

The concept of object boundaries is very important for perceptual grouping of elements (see Figure 2.1).

Additionally it is noted that it is easy for humans to see so called "distractor" objects. "Distractors" are objects that differ from its surrounding elements by one or more features as seen in Figure 2.2. These findings indicate that the process of early vision is guided by automatic detection and processing of simple features. Objects and structures are formed by looking at these features in a preattentive stage. The features are encoded in so call *feature maps*, where each feature is encoded in a separate map. These maps hold the values for the individual features. Elements with the same feature
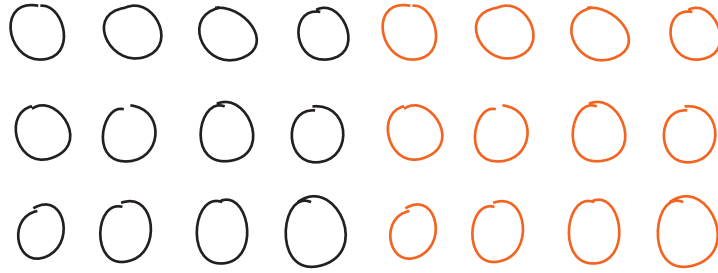
Figure 2.1: The clear boundary between the black and the orange elements is easily perceived by humans and thus an important indicator for perceptual groups.
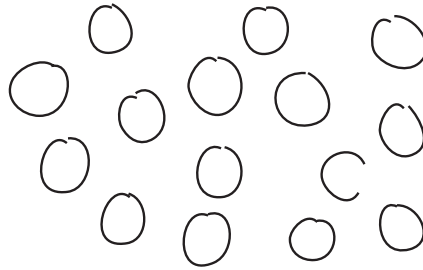
Figure 2.2: Distractor elements in a structure of other elements (like the open circle in this figure) are seen instantaneously (preattentive phase).

(like the same color) are seen as combined objects. This knowledge can be used to find visually important features and use it in the automatic detection of group in sketches and drawings. These features include color, size, contrast, tilt, curvature, and line-endings [41]. Treisman stated that similarity within the feature maps is not linear, it follows the Weber-Fechner law. Additionally Treisman notes that clear boundaries between object structures also make it easier for humans to perceive these structures. Since objects which share features are seen as perceptually grouped in a preattentive process, it can be concluded that for certain simple features, no contextual information is needed for the grouping process. Elements are combined to objects in only a few hundred milliseconds in the human vision system. Although context is an important factor for grouping in sketches, some features likes the ones stated earlier can be seen as universal and contextless. Using these features in Suggero allowed us to discover perceptual groups without knowing the context or domain of a sketch. They are used to find general groups which are perceived in a context agnostic environment.

2.1.3   Perceptual Features

In research of perceptual grouping or perceptual organization, various features have been explored and used. Some features like proximity and continuity can be used as spatial or temporal indicators [8]. There are other features like similarity of color, size or others that can be seen as additional features, often used in a multidimensional feature space. Features like color and size can be extracted as absolute values and used for further computations.

When it comes to grouping, features always stand in relation to each other. Similarity of color for example is a relative measure between two elements, although the feature color is extracted from the single elements. Other features can only emerge if multiple elements are present. To extract a degree of parallelism, multiple (but at least two) elements have to be compared.

As mentioned in literature by for example Treisman [41], the linearity of features is another important factor to consider. Some features like the similarity in size or color are not linear when it comes to human perception. That means that two objects are seen as similar or can be distinguished dependent on different thresholds or variables. Two of the most famous laws for this are the Weber-Fechner law and Stevens' power law. The Weber-Fechner law states that the scale for noticing differences between two stimuli (called the *just-noticeable-difference*) is logarithmic. Stevens' power law also deals with measuring the just-noticeable-difference, but with a different equation. It states that the perceived intensity follows the equation $\psi(I) = kI^a$, where $\psi(I)$ is the subjective, or perceived, magnitude of the particular stimulus. k is a constant for the current scale and the exponent a varies dependent on the type of stimulus (for example for perceiving changes in the visual area $a = 0.5$). These two laws have to be taken into account for the processing of the different features. Especially if the goal of a system is to extract information about similarity or distance of objects, it is important to keep them in mind.

Features like proximity are used throughout different research projects and systems [8, 18, 36, 40], but often no concrete implementation or models are provided. Additionally, systems often use the features as input for genetic algorithms or optimization [18], which makes understanding the use and results of the extracted features even more complex.

2.2   Perceptual Grouping

Thórisson [40] used proximity and similarity to find perceptual element groups and discusses the general usage of such groups for human computer interaction. She mentioned several features that could be analyzed for their similarity like shape, size, color, brightness (similarity in intensity), orientation and texture. For the grouping algorithm in this work, similarity in shape and size and spatial proximity were used to find perceptually salient

groups. Thórisson computed the pairwise relation for all elements and built a graph with the elements as vertices and the relation values as edge weights. Afterwards, minimum cuts were found by using fixed threshold values. The resulting subgraphs were the groups found by the algorithm.

Igarashi et al. [18] used proximity and regularity (in layout continuity) to find structures in card stacks. They extracted various parameters from a linkage model of the input elements. The model was target of automatic parameter tuning with genetic algorithms. With this learning approach from user input, Igarashi et al. tried to find the correct parameters to extract structures like lists and tables from the input. The results were then used for further interactions.

Similarly, Shipman et al. [38] wanted to find element structures like lists or tables for their pen-based whiteboard system. Their editor relied on user selection and gestures to perform selections and interact with them, so no features except user selection were extracted. Gestures made by users were combined with cleanup operations to match alignments of the input. The user-created selections were specified as groups, which were used for interactions like translation or insertion into existing tables. Besides user selection, Shipman et al. introduced the concept of borders to delimit certain structures. Borders were also used as user operations, but could also be seen as a concept worth looking into for perceptual grouping systems like Suggero. The concept of border is closely related to Treisman and Gelade's concept of clear object boundaries [43].

Rome [36] discussed several features like proximity and similarity. In the system named EPICT, the features from Feature Integration Theory were extracted for later analysis. They used different element attributes like fill, color and position to find perceptual groups. Similar elements (based on the extracted features) were put in similarity maps. Each map was then analyzed in terms of proximity and continuity. The resulting groups were united if possible (similar maps were combined), ordered and presented as results.

Saund et al. [37] extracted line art and blobs from their input images. Besides using user selection as a feature, they analyzed the line art and introduced search for three different kinds of paths, including closure, as indications of groupings. The use of closure in terms of object grouping can be connected to Palmer's work on common regions [31]. A common region is an element that surrounds other elements and thus, is perceived as a kind of container for these elements. He stated common regions are an important principle of perceptual grouping, which emphasizes object relationships. The concept of common regions is illustrated in Figure 2.3.

Cates [8] focused on sketch analysis but also showed important features to identify objects. These features have also been used in perceptual grouping. Three main aspects of a sketch were identified, namely spatial, temporal and conceptual relations, which were used for the analysis. She used the concept
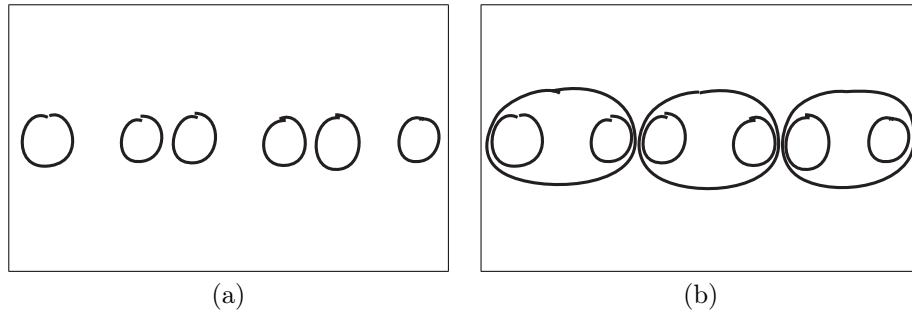
Figure 2.3: The elements in (a) are perceived as pairwise grouped because of their proximity. By adding the ellipses in (b), the prior groupings break in favor of elements sharing a *common region*.

of parallelism and verticality for her analysis, which are also strong features when analyzing sketches.

Finally, Dehmeshki and Stuerzlinger [9] focused on supporting users when selecting perceptual groups. Their work focused on finding element groupings based on proximity and path continuity. They analyzed the movement of selection gestures and the drawings. By extracting the movement and curvature of a selection gesture, they were able to convert the gesture into a measure for distance and continuity of the intended selection. To match selection and input elements, a element graph was constructed with information about curve alignments of the elements. After construction of the graph, the information was matched with the drawing, with the goal to select perceptual groups quickly.

Chapter 3

Concepts

This chapter describes the concepts for finding perceptual groups in Suggero. The general workflow is described as well as the particular concepts to extract particular features. The process of dynamically computing perceptual groups is presented. Finally, the presentation and user interaction with the computed perceptual groups are explained.

3.1   General Workflow

There are different approaches for the workflow of perceptual grouping. Several works, including Suggero, use a multi-phase approach [18, 36, 38, 40]. In a first phase, the *Feature Extraction*, the different features of a digital sketch are computed. In this phase, the elements of a sketch are analyzed and the element relations for features like proximity or similarity are computed. Depending on the input of a system, a prior *Preprocessing Phase* may be necessary before the Feature Extraction Phase. If the input is for example a scan of an image, the strokes have to be extracted, including their attributes like color or thickness. Other systems, including Suggero, use digital input from the mouse or a digital pen. This input is most often processed to single elements (or strokes), which can easily be distinguished and where attributes like color and thickness are set by users before drawing. Suggero also computes different attributes of each element, like its bounding circle or Fourier descriptors and preserves them for later analysis and faster computation. Calculations like these are often computational expensive, so preserving this information instead of computing it every time can result in an increase in performance. After the Feature Extraction, the calculated values are processed in the following *Dynamic Grouping Phase*. In this phase, the perceptual groups, which are used for later presentation and interaction, are found.

3.2   Pre-processing

Since Suggero was created to work with hand-drawn sketches, the input strokes are collections of 2D points (polylines). In the sketching application, users can change both the color and the thickness for each stroke on the fly. Each stroke is pre-processed immediately after it is drawn.

Segmentation and Normalization: The input strokes from the application are not equally sampled. The distance between the 2D points strongly depends on the movement speed of the input device (mouse or pen) while drawing. To implement the similarity and parallelism feature extraction, described below, it is necessary to achieve an equally-sampled representation of each stroke. Therefore, the stroke is re-sampled accordingly (see Figure 3.1). Additionally, a normalized representation is computed and used in the later Feature Extraction.
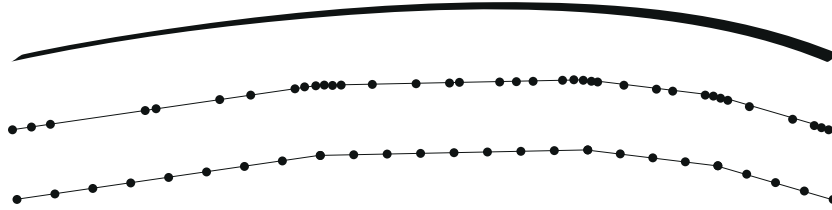
Figure 3.1: The smoothed input stroke as seen by users (*top*), the raw sampling from the device (*middle*), and the re-sampled points (*bottom*).

3.3   Feature Extraction

In Suggero, sketches are analyzed for different features that represent the perceptual relations between drawing elements. These features were selected based on the findings from the Gestalt principles of grouping and the Feature Integration Theory, discussed in Section 2.1.2. To extract a feature, affinity values are computed for all pairs of elements. These affinity values are normalized values ranging from 0 to 1 that express the relation between two elements, where 0 means no relation between elements and 1 stands for highly related elements. The affinity values are used later for Dynamic Grouping. This section describes the choice of features in Suggero and their concepts.

3.3.1   Feature Choice in Suggero

A combination of features from psychological research [10, 31, 33, 41–44] is used for the Feature Extraction. Proximity and similarity are the primary features used, as they are key features identified in Gestalt theory; they have also been successfully applied in previous research projects [9, 18, 40]. In addition to proximity, connectedness is used as an visual important feature [33]. The Feature Integration Theory states that similarity of shape and color are critical features for human perception [41–43]. Given the application context, similarity of thickness is also analyzed, since it is a strong visual feature in a sketch. In addition to these features from the perceptual psychology literature, parallelism is included as it has also been identified as a strong perceptual feature [8].

Relative Element Relations

Whether a perceptual relation between elements exists or not typically depends on the whole content of a sketch. For example, two drawing elements that were original perceived to be related due to their spatial proximity may be perceived as being unrelated after another element is added closer to one than the other (see Figure 3.2). In Suggero, only relative affinity values are computed, expressing the relations between elements.
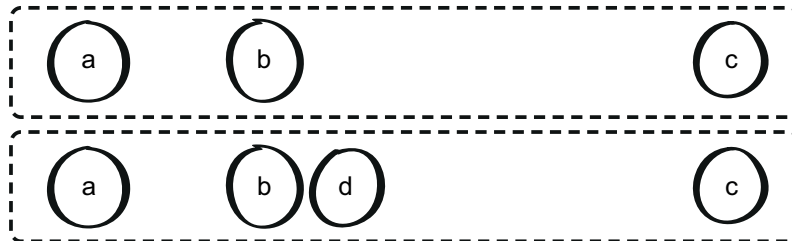
Figure 3.2: The circles a and b are perceived as a belonging together due to their distance to c (*top*). Once circle d is added, this connection breaks and b and d belong together (*bottom*).

3.3.2   Proximity

A main feature of Suggero is *proximity*. Since spatial proximity is a relative measurement, that changes with each element added to a sketch, Suggero uses two different measures for proximity, *global proximity* and *local proximity*.

Global Proximity

Global proximity refers to the distance between two objects, which also considers the length and shape of an element. Two elements can be spatially close on one side and more distant on the other side because of their arbitrary shape and positioning. Therefore, measuring the distance for only one point (e.g. center point) or the endpoint distance would not be sufficient (see Figure 3.3, left, middle). For more accurate measurement, the average-distance of ten points is used for global proximity (see Figure 3.3, right).
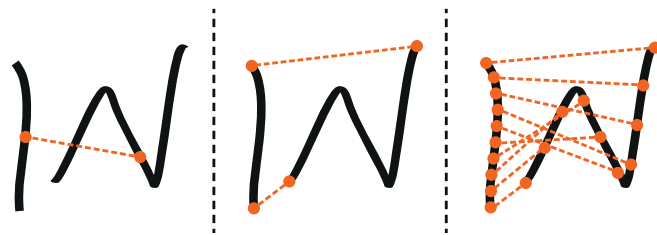
Figure 3.3: Measuring only one or two distances (*left* and *middle*) to analyze proximity would often not be sufficient. Suggero measures the distance of ten different points (*right*) to calculate the global proximity.

Local Proximity

Another measurement for the proximity of two elements that enforces the spatial proximity between two objects is local proximity. Instead of calculating the relation for two elements according to their stroke geometry, the bounding circle is calculated and used to normalize the distance between two elements. The distance is calculated by using the Euclidian distance between the bounding circle centers. By using this distance measure in the later dynamic grouping, elements that clearly belong together based on their bounding circles are weighted stronger. As Figure 3.4 shows, local proximity enforces structures like enclosed elements and containers.

Figure 3.4: The global proximity distance (*left*) is larger than the local proximity distance (*right*). Since the caret (blue) is surrounded by the green circle, the local proximity provides a better match.

3.3.3 Connectedness

Although not mentioned explicitly in the Gestalt theory, connectedness is considered a strong visual feature [33]. Suggero finds intersections between drawing elements and takes this connection into account as features. Two elements can be perceived as connected, because they ei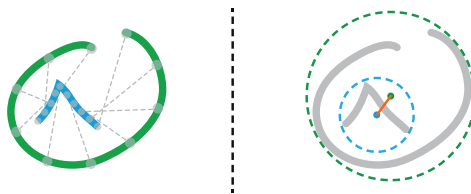ther intersect or because their endpoints are connected. Endpoints of elements do not necessarily need to have a real physical connection to be perceived as connected (see Figure 3.5). To overcome this issue, Suggero uses a different approach to measure the degree of connectedness at endpoints instead of searching for actual connections. The degree of connectedness of two end-points is com-
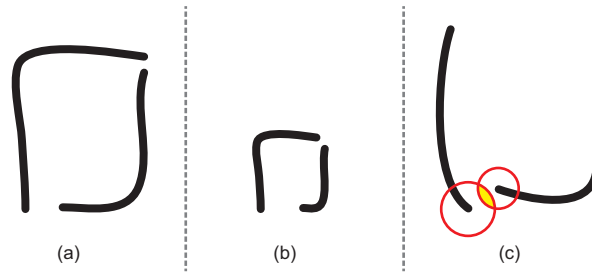
Figure 3.5: Endpoint connectedness depends on the size and distance of objects. Although the distance of bottom left endpoints of (a) and (b) is the same, (a) is more likely to be perceived connected. We calculated so-called tolerance zones (c). The overlap (yellow area) indicates the degree of connectedness.

puted by using tolerance zones [39]. Tolerance zones are circular areas at the end-point of a stroke, used to calculate the degree of connectedness. They take the elements' size as well as the magnitude into account. The overlap of two element's tolerance zones is a good measurement for the degree of endpoint connectedness. The size of the tolerance zones depends on the size of the element and its distance to all other elements in a sketch [39]. This size is determined by calculating the average distance of a particular endpoint to all points of all other strokes in a sketch. This calculation can be computationally expensive; thus, Shpitalni and Lipson's [39] original algorithm was modified to determine the size by using the average distance of the endpoint to the other points of the stroke and the average distance to other tolerance zones.

3.3.4 Similarity

One of the most important features in previous research on perceptual grouping [8, 36, 40, 45] and perceptual psychology [10, 41–44] is similarity. This feature can be used in different dimensions and with different properties of the elements. Suggero takes similarity of *shape*, *(outline) color*, and *stroke*

thickness into account. This list could easily be extended and the system could be adapted if another property is found to be important, for example fill or texture.

Shape

Shape is a very important feature for human perception [10, 41–44]. In Suggero, the pairwise element similarity is computed and used as a feature. Since only single strokes are compared, element similarity in terms of shape for multi-stroke objects is not computed. This would be an important addition for further extensions of the system.

Color

The color similarity of elements is included as a feature in Similarity. Since the systems purpose is to support users in creating digital sketches, only the stroke outlines are used to compare the color of strokes. This could easily extended by differentiating between outline and fill color or by using it as a combined measure for color. The perception of color, as with most other features, is non-linear. This is important to know for any grouping algorithm that needs a binary decision if two objects are similar or not. Suggero uses the Euclidian distance in the CIELAB color space to compute the pairwise similarity of color of the drawing elements. This color space was used because it represents human perception of color differences best. The implementation is described in detail in Chapter 4.

Thickness

Stroke thickness can be seen as another example parameter that is used in Suggero. The current sketching environment provides users with the possibility to draw elements with different stroke thickness. This often leads to users assigning different element thicknesses to indicate groupings and relations. Due to this fact, thickness is used as a feature for Suggero in this application context.

3.3.5 Parallelism

Parallel structures of sketched elements are often non-accidental and can be easily perceived by humans [8]. Parallel lines are perceived as such because they have the same angle and no intersections. The difference in angle is a good measurement for the degree of parallelism. No difference means that the lines are parallel and a difference of 90° means that the lines are perpendicular. The same can be applied to arbitrary strokes (non-straight lines). Besides the difference in angle, the similarity of shapes is important for perceiving objects as parallel (see Figure 3.6).
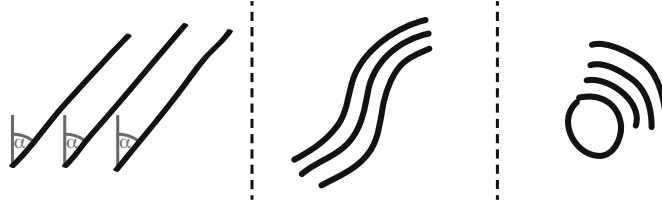
Figure 3.6: Parallelism is perceived due to common orientation (*left*) and similarity (*middle*). Sub-segments are also perceived as parallel (*right*).

Additionally, an element can be parallel to a sub-segment of another element. In Suggero, the pairwise degree of parallelism is computed for all elements by combining the difference in rotation and the similarity. Cates [8] categorizes parallelism under the term of singularities, together with verticality. Verticality is currently not analyzed in Suggero, but would be an interesting feature to investigate.

Summarizing, *Proximity*, *Connectedness*, *Similarity*, *Parallelism* and *Similarity of shape*, *color* and *stroke thickness* are used in Suggero as features for finding perceptually related elements. This list could be extended with other perceptual properties, since only the relative affinity value for a feature needs to be calculated and added as input to the Dynamic Grouping phase. The output of the Feature Extraction is a collection of pairwise affinity values for all elements based on all analyzed features. The normalized values of all these features are then used to calculate the next step, the Dynamic Grouping.

3.4   Dynamic Grouping

The groups we perceive in sketches change with every element that is added, removed, or modified. In the dynamic grouping phase, the output from the Feature Extraction phase is processed and the perceptual groups are computed.

Similarity-based Grouping

Elements are grouped based on their perceptual relation. A hierarchical grouping approach is used, starting with the element structures followed by the individual elements [19]. In Suggero, Dynamic Grouping starts by finding two elements with the closest relation and assigning them to a group. In the next step, the next closest element pairs are found and assigned to a new group and so on. Once created, a group of elements is treated like regular elements; that is, pairwise relations are calculated between the remaining (un-grouped) elements and existing groups. If the closest pair is an element
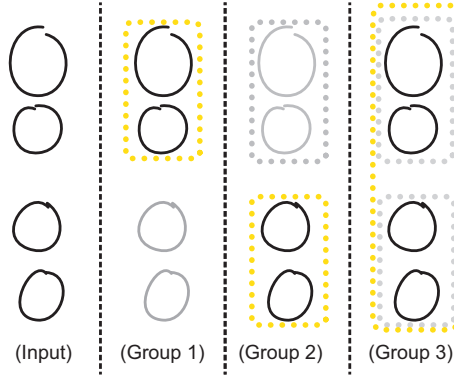
Figure 3.7: Illustration of the Suggero's dynamic grouping process. The closest element pair are grouped first (*Group 1*), followed by the next pair (*Group 2*). The last pair is the two groups themselves, which are assigned to *Group 3*.

and a group, the element is added to the group. This process is continued until all elements are grouped accordingly (see Figure 3.7). All identified groups are stored for later ranking. This grouping method was primarily influenced by a clustering technique, called Hierarchical Agglomerative Clustering (= *HAC*) [27]. The *HAC* technique meets the requirements of grouping the elements based on their relative similarity rather than using fixed thresholds or other rigid methods. Additionally, it is possible to calculate a group-quality value (confidence value) for every group.

Feature Combination

In Suggero, the affinity values from multiple features are used; thus, the input for *HAC* has to be pre-processed. The outputs from the feature extraction phase are multiple matrices with affinity values (one for each feature). Two different strategies of feature combination are used to combine these matrices. The first strategy combines the values from all features on one single matrix by using a weighted sum. The resulting groups after the grouping process can be referred to as *combined feature groups*. The weights are empirically determined and tweaked after the preliminary study. This strategy produces more perceptually complex groups. Since features like shape similarity and parallelism are combined, it is difficult to tell based on which features the groups were formed. The second strategy processes each feature separately with *HAC*, so the inputs are the previously calculated affinity values. The computed groups are later ranked and combined if they contain the same elements. This produces more specialized groups, since the groups are based on particular features (e.g. color) and do not combine multiple features. These groups can be referred to as *specialized feature groups*.

Ranking Groups

In order to provide users with the best groups only, the groups have to be ranked based on their quality (see Figure 3.8). In Suggero, the confidence of groups (ranging from 0 to 1) is computed by averaging the affinity values of all its elements. Additionally, a penalty function, applied to the number of elements in a group, decreases the confidence value for larger groups. This behavior is a subject of the preliminary study.
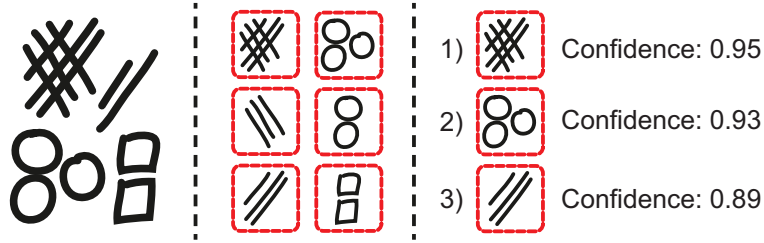

Figure 3.8: Input sketch for the dynamic grouping (*left*), found groups (*middle*) and groups ranked by confidence value (*right*). Confidence values are normalized and range from 0 to 1.

3.5   Interaction

In Suggero, the decision which groups are presented to users is based users' manual selections. After an element is selected manually, Suggero searches in its previously computed perceptual groups for ones that contain this element. After these groups are found, they are ranked based on their confidence value and presented as suggestions. Additionally, users can select multiple elements, which leads to Suggero searching for groups with all those elements. More selected elements gives Suggero more information and results in more exact suggestions. We implemented two different visualizations of the suggestions. The first visualization technique is a simple vertical list and the second visualization is a marking menu [22, 23]. Both show the three top-ranked suggestions.

3.5.1   Suggestions List

The found suggestions of perceptual groups are presented to users with a vertical list. Suggero presents the suggestions with the highest confidence value (best selection group on top) in a vertical list next to the last selected element. Each suggestion shows the elements of the underlying perceptual group. All elements are scaled to fit the area of the suggestion (70 × 70 pixels) and have the same colors as in the sketch to easily identify them in the sketch. Suggestions can be selected by tapping. Suggero then selects all

(a)

(b)

(c)

(d)
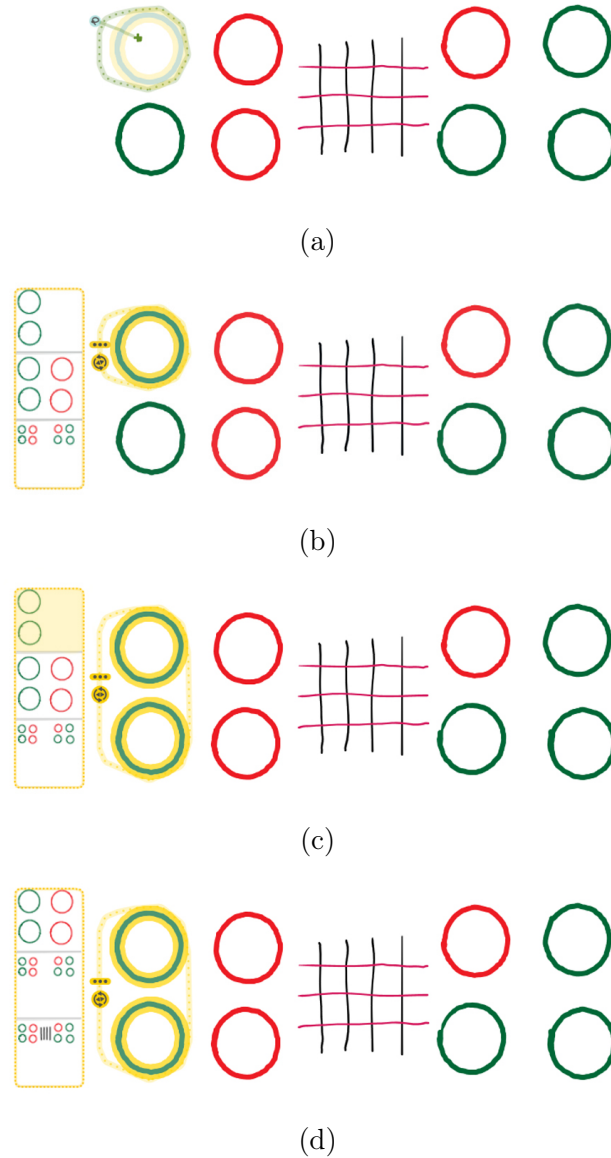
Figure 3.9: After selecting the green circle manually (a), Suggero provides suggestions in form of a vertical list (b). When a suggestion is confirmed by tapping, Suggero selects all containing elements (c). Afterwards, the suggestions are updated to match the new elements (d).

the elements contained in the suggestion (see Figure 3.9). Afterwards, the suggestions are updated based on the newly selected elements. Users can tap though the different levels of suggestions with always increasing group size to enable fast selection of small and large groups.

3.5.2   Marking Menu

A marking menu containing the suggestions was implemented in addition to the suggestions list. The marking menu, developed by Kurtenbach and Buxton [22, 23] is a variation of the pie menu [7]. The suggestions are presented in a radial way (depending on the last pen motion), as shown in Figure 3.10. Additionally, a cancel item is shown to discard the menu. Suggestions are selected when the pen stroke touches them. The menu becomes visible when the pen stops moving (with a 5 pixels threshold) for a fixed time (300 ms).
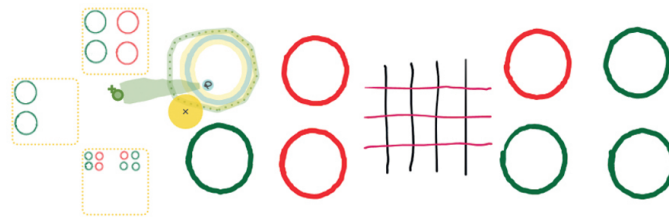
Figure 3.10: As an alternative to the vertical list, a marking menu showing the top-ranked suggestions was implemented.

3.5.3   Manual Selection: Harpoon

For manual selections with Suggero, the *Harpoon selection tool* is used. It was developed after the ideas of Leitner and Haller [24]. The Harpoon selection method is a speed-dependent crossing technique that supports users in performing complex selections. Every time an element gets touched with the Harpoon tool, it is either selected or deselected, depending on its prior state. The faster the stylus is moved when using the Harpoon tool, the bigger the selection area gets (and the more elements are selected). This enables both very accurate and large selections. Figure 3.11 illustrates this behavior.
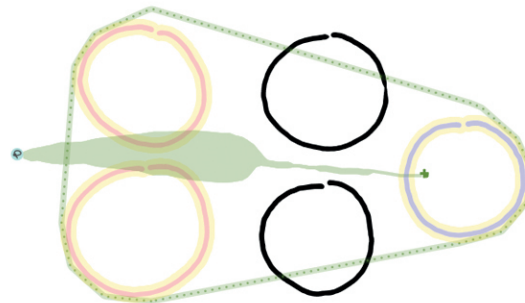
Figure 3.11: The Harpoon tool is speed dependent. At the beginning (left side), to selection area (green) is larger because of faster movement speed.

If users lift the pen in between two selections, the selections are not reseted but preserved and added to any further selection. Removing strokes is possible by simply crossing an already selected stroke again or by clearing the selection with single touching in an area of the application where no stroke is located. These possibilities to simply add and remove strokes from a selection qualify it as the manual selection tool for Suggero.

3.5.4 Sketching application

Suggero is implemented in a sketching application as a selection tool. The sketching application allows users to draw freely and modify the drawing elements before and after their creation. To change color and thickness of an element, the desired properties have to be set before drawing using a menu (see Figure 3.12).
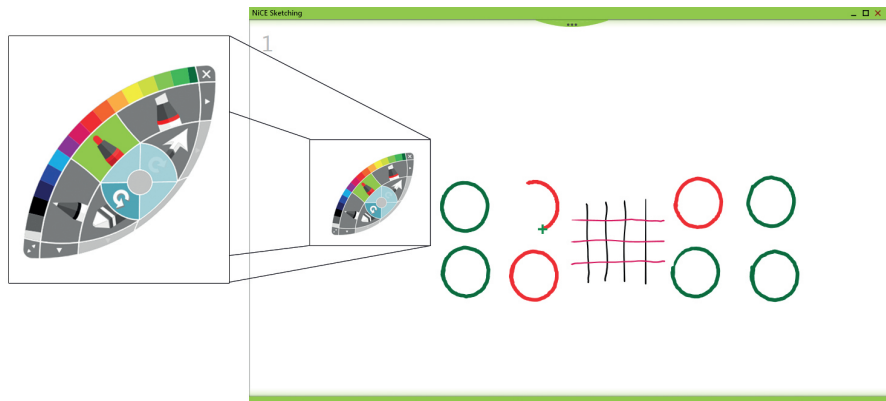
Figure 3.12: The application in which Suggero is embedded. Left is a detail view of the menu. Users can change color and thickness of the elements they draw in advance. Additionally, eraser and selection tools are available.

In this menu, users can also switch to the Eraser tool to remove elements or to the selection tool, which is Suggero. The selected elements can be scaled and rotated by using a scale-rotate handle and recolored with a context menu. The implementation of the sketching application was not part of this thesis. The sketching application can run on regular desktop computers as well as on interactive whiteboards.

Chapter 4

Implementation

The concepts and interactions were explained in Chapter 3. This chapter presents details on the implementation of these concepts used in Suggero. The implementation of the three phases used in the process of finding perceptual groups, *Pre-processing*, *Feature Extraction*, and *Dynamic Grouping* is described, as well as details on methods and tweaks used to improve the performance.

4.1 Pre-processing

To prepare the input of the sketching application (shown in Chapter 3.5.4) for the Feature Extraction and the Dynamic Grouping, it has to be pre-processed. The elements drawn by users are strokes, which are collections of 2D coordinates. Additional information such as stroke thickness and color is provided by the system. No additional work like stroke or edge detection is required, which is an advantage compared to systems where this was necessary (e.g. [37]) due to different input (e.g. scanned images). Pre-processing is needed because the input strokes are un-evenly sampled (coordinates have different distances) and additional element properties have to be computed.

4.1.1 Stroke Properties

A large amount of research has been done on analyzing strokes drawn by users. Primitive recognition research used different properties of drawn strokes like its curvature, direction and speed to find out, what kind of object was drawn [14, 29, 34]. By knowing the drawing objects (for example circles, squares or arrows), the context of the drawing can be identified or information can be added to the objects. Although this is important in the context of sketch analysis, it is not necessarily mandatory knowledge for perceptual grouping systems like Suggero. As discussed earlier in Chapter 2.1, humans don't need context to see perceptual groups. On the other hand, the informa-

tion can be useful to identify important element like text or group separators. Also, being able to compute shape similarity on a higher level by comparing shapes by their type (circles or rectangles) can be useful for perceptual grouping systems.

In the context of this thesis, the coordinates, color, and thickness of strokes are stored for later processing. Additionally, a 2D bounding circle is calculated for all elements, with the center point c being calculated from the average of all coordinates and the radius r being the maximum distance between c and the points of the element's stroke. The bounding circle is used during the later Feature Extraction for calculating proximity.

4.1.2   Segmentation

In order to get an evenly sampled representation of the input strokes, a segmentation algorithm has been applied. The segmentation algorithm from Burger and Burge [6] was modified to achieve this. The original algorithm segments a stroke to a fixed number of points. With this algorithm the points are evenly distributed, but since the input strokes have different lengths, the sample distance is different for each element. For the later Feature Extraction, it was more important to get equi-distant points and less important to get the same number of points for every stroke. All strokes were sampled to have a point distance of 2 pixels. The algorithms from [6] needs the number of sample points N of the output stroke as parameter. In Suggero, the number is calculated dynamically by calculating the overall length of the stroke and dividing this number by the desired point distance (2 pixels)[1].

4.1.3   Normalization

For the calculation of shape similarity, a normalized representation of the input stroke is calculated. The same algorithm as for the stroke segmentation is used (base algorithm from Burger and Burge [6]). In contrary to the segmentation, the normalized representation is calculated with a fixed number of points (100 points) and a fixed distance between the points (1 pixel). The original algorithm is applied to a copy of the input stroke. The result is a stroke consisting of 100 equi-distant points. To achieve a distance of 1 pixel between all stroke points, the stroke is scaled (see Algorithm 4.1).

4.1.4   Fourier Descriptors

For the later calculation of element similarity, Fourier descriptors of each stroke are calculated. Fourier descriptors are a representation of the element's shape in the spectral space [46]. The shape signature of each stroke

---

[1]If the stroke length is not a multiple of 2, the result with have a remainder. In this case, the resulting number of points is rounded down.

Algorithm 4.1: Stroke scaled to all points having the same distance.

1: $\textsc{ScaleStroke}(s, d)$
      Scales a stroke s all points having a pairwise distance d.
      Returns a scale stroke t.
      $s = (p_0, p_1, \ldots, p_{N-1})$ $\triangleright\ p_i \in \mathbb{R}^2,\ \langle x_i, y_i \rangle$
2:      $l \leftarrow \sum\limits_{i=0}^{N-2} \|p_{i+1} - p_i\|$ $\triangleright$ calculate stroke length l
3:      $f \leftarrow \frac{N \cdot d}{l}$ $\triangleright$ calculate scale factor f
4:      $t \leftarrow [\ ]$ $\triangleright$ initialize empty stroke t
5:      $t_0 \leftarrow s_0$ $\triangleright$ add start point of input stroke to t
6:      **for** $i \leftarrow 1 \ldots N - 1$ **do**
7:          $\delta \leftarrow p_{i-1} - p_i$
8:          $t_i \leftarrow p_{i-1} + \delta \cdot f$ $\triangleright$ add point with scaled distance
9:      **end for**
10:    **return** t.
11: **end**

---

is calculated using a commonly used algorithm (as described in [46]) that extracts complex coordinates from the shape. Since the Fourier descriptors need a continuous shape, each stroke is considered periodic by repeating the shape. For a stroke with N points this means that $p_{(N+i)} = p_{((N+i) \bmod N)}$. To avoid problems with the descriptors, the normalized representation is used to calculate the descriptors. As a compromise between accuracy and performance, 30 Fourier Descriptors are calculated for each stroke. The descriptors are used in the later Feature Extraction to compare elements in terms of shape similarity.

4.2   Feature Extraction

In this section, the different algorithms to extract the features *proximity*, *similarity*, *connectedness* and *parallelism* are explained. The concepts and choice of features were presented in Chapter 3.3.

4.2.1   Proximity

As explained earlier in Section 3.3.2, Suggero uses two different measures of connectedness. *Global proximity* computes the distance relation between the elements and takes the elements' shape and size into account. *Local proximity* calculates the element relation with the help of the bounding circle. This enforces structures like enclosed elements.

Global Proximity

Global proximity is calculated for each pair of elements in a sketch. The distance between 10 points is calculated and the average is used as a measure for the distance. This distance is normalized to the greatest distance of elements in a sketch. This way, the relativeness of element relations (described in Section 3.3.1) is taken into account. Figure 3.3 illustrates this algorithm. Since each element has two end-points, the correspondence of the points used in the distance calculation is unclear. To overcome this issue, the nearest pair of endpoints of two elements is used as start points and the other distances are calculated successively. Algorithm 4.2 shows the calculation of the global proximity for a pair of elements.

Algorithm 4.2: Calculates the global proximity between two elements.

1: $\textsc{GlobalProximity}(s_1, s_2, t)$
     Calculates the distance between the two elements s_1 and s_2.
     t is the number of points to check.
     Returns the distance d.
     $s_1 = (p_{1,0}, p_{1,1}, \ldots, p_{1,N-1})$ $\triangleright$ $p_{1,i} \in \mathbb{R}^2$, $\langle x_{1,i}, y_{1,i} \rangle$
     $s_2 = (p_{2,0}, p_{2,1}, \ldots, p_{2,M-1})$ $\triangleright$ $p_{2,i} \in \mathbb{R}^2$, $\langle x_{2,i}, y_{2,i} \rangle$
2:     $calcReverse \leftarrow \|s_{1,0} - s_{2,0}\| > \|s_{1,N-1} - s_{2,M-1}\|$
                                         $\triangleright$ check if correspondence is reversed
3:     $\delta_1 \leftarrow \lfloor \frac{N}{t} \rfloor$ $\triangleright$ index offset for each step of t
4:     $\delta_2 \leftarrow \lfloor \frac{M}{t} \rfloor$
5:     $d \leftarrow 0$ $\triangleright$ initialize distance d
6:     **for** $i \leftarrow 0 \ldots t$ **do**
7:        **if** calcReverse **then**
8:           $d \leftarrow d + \|p_{1,N-i\cdot\delta_1} - p_{2,M-i\cdot\delta_2}\|$
9:        **else**
10:          $d \leftarrow d + \|p_{1,i\cdot\delta_1} - p_{2,i\cdot\delta_2}\|$
11:        **end if**
12:     **end for**
13:     **return** d.
14: **end**

Local Proximity

In addition to global proximity, the pairwise local proximity is calculated for all elements. In the Pre-processing, the bounding circles for all elements are calculated. These are used for computation of local proximity. The distance of two strokes s_1 and s_2 is calculated by calculating the Euclidian distance d of their bounding circle centers c_1 and c_2, which is $d \leftarrow \|c_1 - c_2\|$.

The average radius r of the bounding circles' radii (r_1 and r_2) is calculated with $r \leftarrow \frac{r_1 + r_2}{2}$ and used to normalize the calculated distance to get the

affinity value. The resulting value a is calculated as $a \leftarrow \frac{d}{r}$. The value of a indicates the spatial relation between the two elements ranging from 0 to 1, where 1 means overlapping centers and a smaller value means less relation in terms of local proximity.

4.2.2   Connectedness

As stated in Section 3.3.3, connectedness is considered a strong visual feature for perceptual groups [33]. In order to calculate corresponding affinity values indicating the degree of connectedness between the elements, two different measures are used. The first, the *element connectedness*, searches for intersection between the element stokes. This is implemented in Suggero by using a line-line intersection algorithm. All pairs of points for two elements are checked for intersections[2]. To avoid too much computational effort, only elements with overlapping bounding circles are checked.

The second measurement, the *endpoint connectedness* searches for connected endpoints. This is implemented in Suggero using so called *tolerance zones*.

Endpoint Connectivity

To measure the connectedness of two strokes at their endpoints, it is not feasible to use fixed constraints like a constant distance threshold. Shpitalni et al. [39] proposed an algorithm to find endpoint connections depending on the distance and magnitude of all strokes. This algorithm calculates so called tolerance zones for each endpoint of a stroke and looks for connections between two endpoints by checking the corresponding tolerance zones for overlaps (see Figure 4.1). The tolerance zones have different sizes depending on the stroke length and are circular.
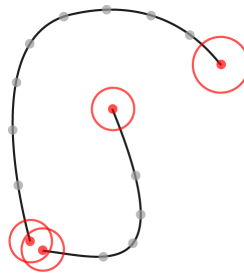

Figure 4.1: The bottom tolerance zones overlap, indicating a high degree of endpoint connectedness.

---

[2]Line-line intersection algorithm on Mathworld http://mathworld.wolfram.com/Line-LineIntersection.html

Although this algorithm has advantages in terms of precision and relationship correspondence of strokes, it has some disadvantages. Equi-distant sampling points are needed in order to achieve a valid calculation of the of a stroke's tolerance zones. Otherwise the calculation leads to different tolerance zones sizes for the same stroke, as seen in Figure 4.2.
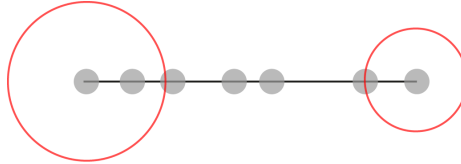

Figure 4.2: Non-equidistant sampling creates tolerance zones with different sizes.

Due to the fact that the algorithm computes the tolerance zone radius relative to the distance of all strokes (and all of its stroke points), it is computational expensive. To provide a fast and still precise measure of endpoint connectedness, a *modified tolerance zone algorithm* is used.

Modified Tolerance Zone Algorithm

Instead of calculating the average distance of an endpoint to all other existing stroke points, only the internal tolerance zone radius is calculated. By averaging the distances from the current stroke's endpoint to the other points, the radius of the tolerance zone is calculated (see Figure 4.3). The center is the current endpoint (see Algorithm 4.3).
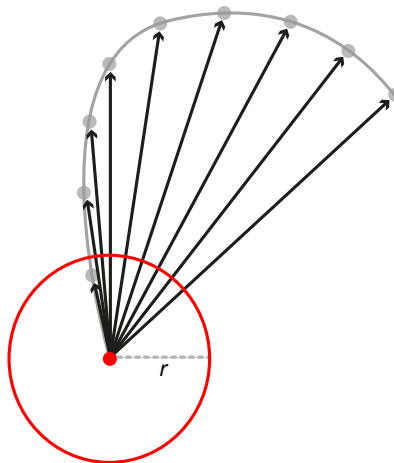

Figure 4.3: The radius r is the average distance from the stroke's endpoint to all other points.

Algorithm 4.3: Tolerance Zone calculation for a stroke

1: $\textsc{CalculateToleranceZone}(S, \omega)$

  Calculates the tolerance zones for a list of strokes S. ω is the size of the tolerance zone (e.g. 2). Returns tolerance zone list T.
  $S = (s_0, s_1, \ldots, s_{N-1})$
  $s_k = (p_{k,0}, p_{k,1}, \ldots, p_{k,M-1})$ $\hspace{2cm}$ ▷ $p_{i,j} \in \mathbb{R}^2$

2: $\hspace{1cm} T \leftarrow [\ ]$ $\hspace{1cm}$ ▷ initialize empty list of Tolerance Zones, Size is $2N$

3: $\hspace{1cm}$ **for** $i \leftarrow 0 \ldots N-1$ **do**

4: $\hspace{2cm} s \leftarrow S_i$ $\hspace{1cm}$ ▷ $s\langle t_0, t_{M-1} \rangle$, s has tolerance zone at p_0 and p_{M-1}
  $\hspace{3.5cm}$ ▷ $t_j \langle c_j, r_j \rangle$, Tolerance Zone t_j, center c_j, radius r_j

5: $\hspace{2cm} c_0 \leftarrow p_0$

6: $\hspace{2cm} r_0 = \frac{1}{\omega M} \sum_{j=0}^{M-1} \|p_j - c_0\|$

7: $\hspace{2cm} c_{M-1} \leftarrow p_{M-1}$

8: $\hspace{2cm} r_{M-1} \leftarrow r_0$ $\hspace{2cm}$ ▷ t_0 and t_{M-1} have same radius

9: $\hspace{2cm} T[2i] \leftarrow t_0$

10: $\hspace{2cm} T[2i+1] \leftarrow t_{M-1}$

11: $\hspace{1cm}$ **end for**

12: $\hspace{1cm}$ **return** T.

13: **end**

This leads to a faster calculation, because only a fraction of point-to-point distance calculations are needed compared to the original algorithm. The algorithm only takes the size of the stroke in account, for which the tolerance zones are calculated. To also include the magnitude between different strokes, the tolerance zone of an endpoint is compared to all other tolerance zones. If two tolerance zones intersect, the smaller diameter of the two tolerance zones is assigned to both (see Figure 4.4).
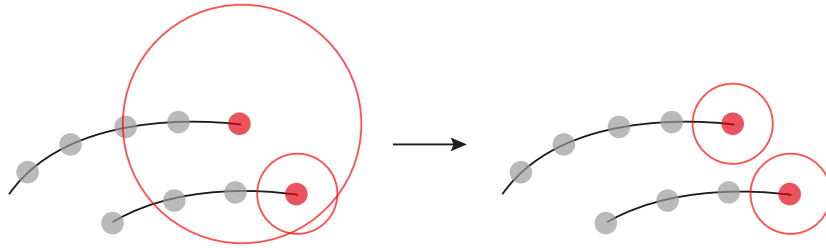

Figure 4.4: Assignment of smaller tolerance zone. This algorithm is used to include the magnitude for differently sized strokes.

This way, not only size but also the distance relation between elements is taken into account (see Algorithm 4.4). Finally a normalized measurement or *degree of endpoint connectedness* (d) is computed by calculating the average

distance c between the endpoints relative to the size of its tolerance zones with the equation $d \leftarrow \frac{\|c_i - c_j\|}{r_i + r_j}$. A smaller number indicates a higher degree of endpoint connectedness. The degree of connectedness is included in the later grouping calculation.

Algorithm 4.4: Update all Tolerance Zones in all strokes

1:	UPDATETOLERANCEZONES(T)
	Updates the tolerance zones in list T based on proximity.
	$T = (t_0, t_1, \ldots, t_{M-1})$ $\triangleright\ t_k \in T, \langle c_k, r_k \rangle$
2:	**for** $i \leftarrow 0 \ldots M-1$ **do**
3:	**for** $j \leftarrow 0 \ldots M-1$ **do**
4:	t_i, t_j $\triangleright\ (t_i, t_j) \in T$
5:	$d = \|c_i - c_j\|$ $\triangleright$ Distance between t_i and t_j
6:	**if** $d < r_i + r_j$ **then** $\triangleright$ Check for intersection
7:	$r_i = \min(r_i, r_j)$ $\triangleright$ Both tolerance zones get smaller radius
8:	$r_j = \min(r_i, r_j)$
9:	**end if**
10:	**end for**
11:	**end for**
12:	**end**

Modified Version versus Original

The advantage of this algorithm is a gain in performance while still preserving a measurement for the magnitude between elements. The original version of the algorithm is more precise for calculating the size relative of the tolerance zones to all other elements, resulting in a better magnitude measure.

For the subject of perceptual grouping the precision of the modified tolerance zone algorithm is good enough, since endpoint connectedness is just one of multiple measures. The modified version of the algorithm also offers a continuous measure of connectedness, while the original version results in a binary decision if two endpoints are connected or not.

4.2.3 Similarity

Pairwise similarity in Suggero is calculated for different element properties. Similarity of shape, color and stroke thickness are computed and considered in the later Dynamic Grouping. The corresponding concepts of Suggero were described in Section 3.3.4.

Shape Similarity

For Suggero, two different measures of shape similarity are implemented. The first approach calculated the affinity values by using a linear least squares

optimization approach. Additionally, a second approach using Fourier descriptors was implemented. In the current version of Suggero, the first approach is used to calculate sub-segment similarity for extracting the feature of parallelism. The second approach is used to calculate the pairwise element similarity. Fourier descriptors are computed in the Pre-processing. This is why the comparison in the Feature Extraction is less computational expensive than the linear least squares optimization. A comparison of the two approaches would be an interesting topic for future research. Both approaches are described in the next paragraphs.

Linear Least Squares Optimization: To find a measurement for the similarity of two strokes s_1 and s_2, a linear least squares optimization method is used. To avoid including scale in the measure, both strokes are resampled to have the same number of equi-distant points (see Section 4.1.2). By normalizing the strokes to a fixed number of points and distance, scale invariance is achieved. After this normalization process, the strokes are assigned to a source matrix A and a target vector b. A transformation vector v is created to get the transformation between the two strokes. The source matrix is computed under the assumption that every point (x, y) of the source stroke can be mapped to every point (x', y') of the target stroke with an affine transformation as in equation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} tx' \\ ty \end{pmatrix}. \tag{4.1}$$

The equations $x' = x.a - y.b + tx$ and $y' = y.a + x.b + ty$ are brought to matrix form to get the source matrix A as

$$A = \begin{pmatrix} x_{1,0} & -y_{1,0} & 1 & 0 \\ y_{1,0} & x_{1,0} & 0 & 1 \\ x_{1,1} & -y_{1,1} & 1 & 0 \\ y_{1,1} & x_{1,1} & 0 & 1 \\ \cdots \\ x_{1,M-1} & -y_{1,M-1} & 1 & 0 \\ y_{1,M-1} & x_{1,M-1} & 0 & 1 \end{pmatrix}. \tag{4.2}$$

a and b are the rotation and scaling (combined) and t_x and t_y the translation of the transformation. The target vector b contains the x and y coordinates of stroke s_2 (x' and y') and is constructed as

$$b = \begin{pmatrix} x_{2,0} \\ y_{2,0} \\ x_{2,1} \\ y_{2,1} \\ \cdots \\ x_{2,M-1} \\ y_{2,M-1} \end{pmatrix}. \tag{4.3}$$

Finally, the transformation vector

$$v = \begin{pmatrix} a \\ b \\ tx \\ ty \end{pmatrix} \tag{4.4}$$

holds the values needed to transform the source stroke to the target stroke. This method solves the equation $A \cdot v = b$ by minimizing the error vector r. Due to the fact that only in a small number of cases the source stroke can be mapped exactly onto the target stroke, the linear least squares optimization method minimizes the error vector. The vector

$$r = \begin{pmatrix} r_{0,x} \\ r_{0,y} \\ r_{1,x} \\ r_{1,y} \\ \dots \\ r_{M-1,x} \\ r_{M-1,y} \end{pmatrix} \tag{4.5}$$

contains the Euclidian distances between the stroke points, which is a direct measure of similarity between two strokes. By not including the transformation vector and only using the point-to-point distance in the similarity measure, it becomes translation and rotation invariant. Algorithm 4.5 show the complete algorithm to compute the similarity measure between two strokes. By applying the transformation to the source stroke, the transformation matches the target stroke as good as possible (see in Figure 4.5).

Algorithm 4.5: Calculates similarity between two strokes

1: CALCULATESIMILARITY(s_1, s_2)
      Calculates similarity of two strokes based on fitting error. Returns an error measure e.
2:     $A \cdot v = r + b$ $\triangleright$ r is error vector.
3:     Solve for v, $\|r\| \to \min$
4:     $e \leftarrow \|r\|^2$
5:     **return** e.
6: **end**

Fourier Descriptors: The Fourier descriptors (calculated earlier) are used in Suggero for pairwise comparison of shapes. They were computed from the normalized representation of the elements, normalized in terms of stroke points and point distance. Each stroke s has a list of descriptors $Z = (z_0, z_1, \dots, z_M - 1)$ (computed in the Pre-processing), with each descriptor being a complex coordinate $z = x(t) + i \cdot y(t)$.
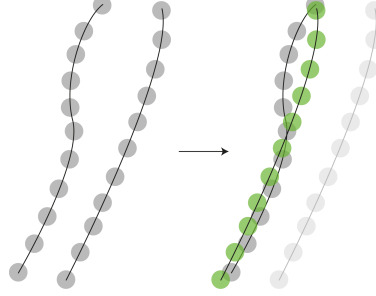
Figure 4.5: Source stroke (right) is transformed to target stroke (*left*).

The first descriptor represents the centroid distance (position), the second represents the element's rotation [46]. Since the position is not a factor for shape similarity, the first descriptor is omitted in the later comparison. To compute the similarity between two elements s_1 and s_2, the descriptors are compared and the average pairwise distance of the descriptors d is calculated as

$$d \leftarrow \frac{1}{N} \sum_{i=0}^{M-1} \|z_{1,i} - z_{2,i}\|. \tag{4.6}$$

The resulting distance is normalized to the length of the longer stroke. The results are affinity values ranging from 0 to 1, where 0 means no similarity and 1 means the elements have an equal appearance. This approach is very sensitive to changes in frequency (sampling of the points), and rotation of the elements. Observations show that the results seem to fit human perception, although most similarities ranged from 0.8 to 1, also for visually very different shapes. Using Fourier descriptors for calculating the similarity values, especially in terms of reliability, needs to be subject of further investigation.

Color

Color is an important visual feature for perceptual groups, as stated earlier in Section 3.3.4 and perceptual research [41–43]. Users have the possibility to change the outline color of elements in the used sketching application. In order to meet the requirements of human perception, the CIELAB color space is used to compare colors. The color space was designed to match the human perception of color distance (see e.g. [5]).The color distance between two elements is calculated using the Euclidian distance (of all components L, a, and b). For two strokes s_1 and s_2, this would mean the distance d is $d \leftarrow \|L_1 - L_2\| + \|a_1 - a_2\| + \|b_1 - b_2\|$. As with global proximity, the color distance is normalized using the greatest pairwise difference in color in a sketch. This way, the result is normalized, ranging from 0 to 1.

Thickness

In the current context of the application, users are able to change the stroke thickness of elements. This way, thickness becomes a good feature to include in the later Dynamic Grouping. The stroke thickness for each element is provided by the application as the average stroke thickness. While drawing, the thickness changes with the movement of the mouse or pen. The thickness is controlled using digital ink, meaning when the pen is moved fast, the stroke gets thinner, and when the pen is moved slow, the stroke gets thicker. This behavior is also provided by the sketching application. Suggero calculates the pairwise difference in thickness using the element's average thickness and normalizes it to the greatest thickness of the sketch. The resulting affinity values for the feature thickness are used in the later Dynamic Grouping.

4.2.4 Parallelism

Another important factor for visual grouping of strokes is parallelism. To compute a single measurement for parallelism, multiple aspects are taken into account, derived from visual key features of parallel strokes. These features are the similarity of two elements, the angle of rotation and the rotated normal distance. It is important to point out that parallelism is not only a feature of two strokes with the same length and that parallelism is not a bidirectional feature. This means that a shorter stroke can be parallel to a sub-segment of a longer stroke without the longer stroke having a parallel connection to the shorter stroke. Due to this fact comparing two non equi-length strokes means comparing the shorter stroke to every sub-segment of the longer stroke. These sub-segments have the same length as the shorter stroke and shift from the start to the end of the longer stroke. Figure 4.6 visualizes the segment shifting.
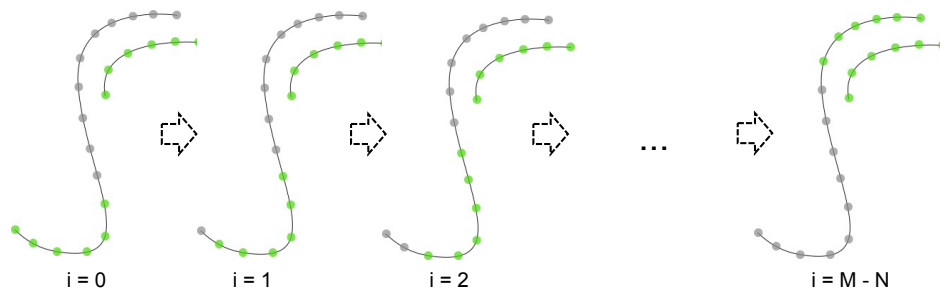

Figure 4.6: Shorter stroke is compared with every subsegment of the longer stroke and the corresponding degree of parallelism is analyzed.

Algorithm 4.6 shows the process of calculating the error measure for every subsegment. If both strokes have the same length, the sub-segment of s_2 is

the whole stroke and the list of error measures E contains only one element.

Algorithm 4.6: Calculates similarity between two strokes

1: $\textsc{CalculateParallelism}(s_1, s_2)$
    Calculates parallelism measure of two non equi-length strokes. Returns a list of error measures E.
    $s_1 = (p_{1,0}, p_{1,1}, \ldots, s_{N-1})$, $s_2 = (p_{2,0}, p_{2,1}, \ldots, s_{M-1})$ $\quad\quad \triangleright M > N$
2: $E = [\ldots] \leftarrow$ empty list of error measures. $\quad \triangleright E = (\sigma_0, \sigma_1, \ldots, \sigma_M)$
3: **for** $i \leftarrow 0 \ldots M - N$ **do**
    Compute error measure for every subsegment.
4: $\quad s_{2,i} = (p_{2,i}, p_{2,i+1}, \ldots, p_{2,i+N})$ $\quad\quad \triangleright s_{1,i}$ is subsegment of s_2
5: $\quad \sigma_i \leftarrow ComputeParallelismMeasure(s_1, s_{2,i})$
6: **end for**
7: **return** E.
8: **end**

The process of finding the measure for parallelism contains several steps to calculate all three error measures, namely fitting error, rotation displacement and rotated normal distance. To compute the fitting error and the rotation displacement, the linear least squares optimization method from computing the similarity (see Section 4.2.3) is used again. The method can be used because the two strokes have equi-distant points and the same number of points because only subsegments are compared. The results are the transformation vector v and the error vector r.

Fitting error: The fitting error is an important measure for parallelism. Two strokes or segments are less likely to be perceived as parallel if the fitting error is too large. The linear least squares optimization outputs the fitting error for all points (vector r) of the two input strokes (see Figure 4.7). With this vector, the error measure e can be computed as $e \leftarrow \|r\|^2$.
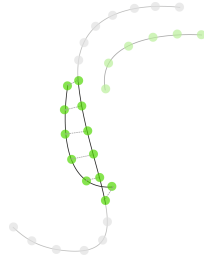

Figure 4.7: The fitting error (Euclidian distance of points) is calculated for every stroke-to-stroke sub-segment match.

Rotation displacement: A large rotation indicates a loss of parallelism as visual feature for the two elements. Therefore, rotation displacement is a good indicator for finding the degree of parallelism between two stokes. The result of the linear least squares optimization is the transformation vector v, which contains the rotation in the first two rows of the vector. With these two factors the rotation can be computed as follows and taken in account for the overall parallelism measure σ. The transformation vector v is calculated as

$$v = \begin{pmatrix} a \\ b \\ tx \\ ty \end{pmatrix},\tag{4.7}$$

while the calculation of the rotation $\alpha \leftarrow \tan^{-1}\frac{b}{a}$ is the arctan of the second and the first element of v.

Rotated normal distance: The rotated normal distance is the distance between the normal of the source stroke rotated by the angle α passing through the first point of the target stroke segment and the first point of the source stroke s_1. In the first step, the normal is computed. Therefore, the fitting line of the source stroke is calculated (see Algorithm 4.7). A total least squares method can be used to compute the fitting line.

Algorithm 4.7: Fitting calculation for a stroke

1:   FITTINGLINE(s)
         Calculates total least squares fitting line L_k of a stroke.
         Stroke $S = (p_0, p_1, \ldots, p_{M-1})$ $\triangleright\ p_{i,j} \in \mathbb{R}, \langle x_{i,j}, y_{i,j} \rangle$
2:       $L\langle A, B, C \rangle, Ax + Bx + C = 0$ $\triangleright\ L$ is algebraic line
3:       Solve for L, $(Ax + By + C) \rightarrow \min$
4:       **return** L.
5: **end**

The resulting fitting line is an algebraic line in the form $L_1 \rightarrow \min(Ax+By+C)$ with the parameter A, B and C. The normal N_1 follows this parameters and can be calculated as

$$N_1 = \begin{pmatrix} A \\ B \end{pmatrix}.\tag{4.8}$$

To rotate the normal, the vector is multiplied by the rotation part (a and b) of the transformation vector v. The rotated normal is then calculated as

$$N_1' = \begin{pmatrix} A \\ B \end{pmatrix} \cdot \begin{pmatrix} a & -b \\ b & a \end{pmatrix}.\tag{4.9}$$

After calculating the rotated normal the last step is to calculate the distance δ between the line going through the first point of the target stroke (segment)

$p_{2,i}$ with the rotated normal as the slope of the line and the first point of the source stroke $p_{1,0}$. Figure 4.8 illustrates the calculation of the line to point distance.
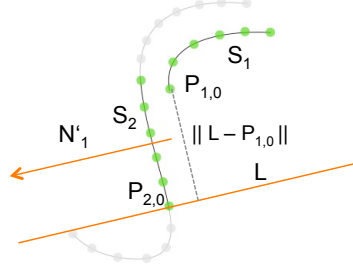

Figure 4.8: After calculating the rotated normal N'_1, the distance $P_{1,0}$ is calculated.

After the computation of the three parallelism measures e, α and δ they are combined to an overall parallelism measure σ. The complete algorithm is listed in Algorithm 4.8.

Algorithm 4.8: Calculates parallelism between two strokes

1: CALCULATEPARALLELISM(s_1, s_1)
   Calculates parallelism measure of two strokes based on fitting error, angle and translation. Returns a parallelism measure σ.
   $s_1 = (p_{1,0}, p_{1,1}, \ldots, p_{1,M-1})$
   $s_2 = (p_{2,0}, p_{2,1}, \ldots, p_{2,M-1})$ ▷ Strokes have same number of equi-distant points
2:    $A \cdot v = r + b$ ▷ r is error vector.
3:    Solve for v, $\|r\| \leftarrow \min$
4:    $e \leftarrow \|r\|^2$
5:    $\alpha \leftarrow \tan^{-1} \frac{b}{a}$
6:    $L_1 = CalculateFittingLine(S_1)$
7:    $N_1 = \begin{pmatrix} A_1 \\ B_1 \end{pmatrix}$ ▷ Normal vector of L_1
8:    $N'_1 = \begin{pmatrix} A'_1 \\ B'_1 \end{pmatrix} = \begin{pmatrix} A_1 \\ B_1 \end{pmatrix} \cdot \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$ ▷ Rotate N_1 by α
9:    $d = PointToLineDistance(p_{1,0}, p_{2,0}, N'_1) \cdot \frac{1}{M}$
      Compute distance between point $p_{1,0}$ and line from $p_{2,0}$ and vector N'_1 and normalize to stroke length
10:   $\sigma = CombineFeatures(e, \alpha, d)$
11:   **return** σ.
12: **end**

4.3   Dynamic Grouping

Finding groups (often referred to as clusters) from the extracted features is not a one-time or static process. With every new input or modification of a sketch the relations between elements change. Two elements that do not seem to belong together can form a group if a third stroke is added. The same way, a grouping can break apart if one particular element is removed.

4.3.1   Input

This section describes the input preparation for the two different types of feature combinations described in Section 3.4. The resulting groups are the *combined feature groups* and the *specialized feature groups.*

Input for Combined Feature Groups

Combined feature groups are calculated with a single affinity value for each stroke pair (value in the distance matrix) for all the dimensions. This way, the number of dimensions used in the later grouping process can be reduced to one, resulting in a simple two dimensional distance matrix. The distance matrix for the combined feature groups is a weighted sum of all features, as seen in Algorithm 4.9. The weights were determined empirically and tuned based on the results of the first experiment. The resulting weights for the features can be found in Table 4.1.

Algorithm 4.9: Process input for combined feature groups

1: $\textsc{CombineFeatures}(F, W)$

Combines the features for all feature matrices F using the weights W.

Weights W are normalized to sum to 1.

Outputs a matrix t of weighted affinity values.

$F = (f_0, f_1, \ldots, f_{N-1})$ $\triangleright f_i =$[M] [M] , $f_i(k,l) \in \mathbb{R}$ holds all affinity values for a specific feature

$W = (w_0, w_1, \ldots, w_{N-1})$ $\triangleright w_i \in \mathbb{R}$, weight for a specific feature

2:      $t =$[ ] [ ] $\triangleright$ initialize empty matrix.

3:      **for** $i \leftarrow 0 \ldots M - 1$ **do**

4:          **for** $j \leftarrow 0 \ldots M - 1$ **do**

5:              $t_{i,j} \leftarrow \sum\limits_{k=0}^{N-1} f_k(i,j) \cdot w_k$ $\triangleright$ sum weighted features

6:          **end for**

7:      **end for**

8:      **return** t.

9: **end**

Table 4.1: Weights for combined and specialized feature groups. Since all features have different scales, the weights cannot be seen under the view of importance for human perception. This applies for both combined and specialized feature groups.

Feature	*Combined*	*Specialized*
Global Proximity	0.05	0.05
Local Proximity	0.10	0.05
Connectedness	0.10	0.25
Endpoint Connectedness	0.05	0.05
Shape*	0.15	0.6
Color*	0.25	0.6
Thickness*	0.20	0.6
Parallelism*	0.10	0.6

Features with (*) are processed separately for the specialized feature groups.

Input for Specialized Feature Groups

Getting groups that are more specialized in specific features than complex combinations of features can result in high quality groups. After the feature extraction phase, each feature is represented by a separate distance matrix. To get the specialized feature groups, all the matrices are processed separately with the dynamic grouping algorithm and combined later in a post-processing step. Although calculation of the groups from separate features can result in good groups, a major problem arises. By treating proximity as a completely separated feature, other features like color and thickness produce groups spread all over the drawing because spatial proximity is no longer included. Proximity is a visually very important feature for most of the groups. To avoid these spatially spread groups, each feature dimension is combined with the extracted feature dimensions for proximity. Every distance matrix for a feature is then a weighted sum of the feature itself and the feature proximity. As with the combined feature groups, the weights were empirically determined based on testing and an experiment. The weights can be found in Table 4.1.

4.3.2 Requirements of the Dynamic Grouping

Finding a good method for grouping strokes of a sketch demands prior knowledge of the requirements. These requirements are based on different aspects of the whole process of perceptual grouping. They emerge from the need to

handle the input correctly as well as computational aspects and with looking at the expected results.

Input Compatibility

The output from the feature extraction phase are values that express the relationships between strokes of a sketch, depending on a specific feature. These values can be normalized. Additionally, the input for the clustering process can be multi-dimensional, where the number of dimensions equals the number of different extracted features. A high number of dimensions can introduce disadvantages. The results of multi-dimensional clustering are often complex and hard to reproduce. Another issue is a loss of precision with an increasing number of dimensions. This issue is called the "Curse of Dimensionality", first mentioned and described by Bellman in 1957 [4]. To overcome these issues, the features can be combined earlier or the dimensions can be reduced. Combining the features means creating a weighted sum for each element relation. This process does not lead to a decreasing precision, but determining the weights can be challenging. A method to reduce the dimensions for the later grouping is the principal component analysis (= PCA). It is efficient for higher dimensional data but can also lead to a loss of precision due to the process of combining dimensions.

Additionally, the grouping method needs to be able to deal with different numbers of elements. On the one hand, it should be able to handle a small number of elements, because groups can be formed between as little as two elements. On the other hand, it must deal with a higher number of elements, because complex drawings easily consist of hundreds of elements. Besides dealing with different element quantities, the system also needs to be able to complete the computation in real time.

Fuzziness

Elements of a sketch often belong to more than one group. A stroke can be perceived as part of a group because the elements share the same color and at the same time belong to another group because the shapes are similar. As Kelley states in [20], drawings are perceived in a hierarchical manner, starting with the groups and ending with the single elements. A grouping algorithm needs to reflect this process. The resulting groups should not be all-or-nothing groups where every element occurs exactly once. The fuzziness in cooperation with a quality measure leads to a good support for outliers, because groups that are formed more or less accidentally (or would be perceived as accidentally) have a low quality.

Quality Measure

Grouping elements in a fuzzy or hierarchical manner can lead to a large number of computed groups. For the later presentation and interaction of these groups, a quality measure for each group is mandatory. Possible methods are calculating the inter- and intra-cluster scatter for each group after the process. A method that includes a quality measure based on the input, the affinity values, is preferable. With a measurement for the quality, it is possible to rank the resulting groups and let the user of the system interact only with the top ranked groups.

Non-parametric

The system never knows the number of groups in advance. Also, there is no way to add this knowledge during the runtime of the application. Having users to specify the number of groups before any interaction is not acceptable and would result in an unusable system. Many classical clustering approaches like kd tree clustering or k-means clustering require a prior knowledge of the number of clusters. K-means clustering starts with an arbitrary partitioning and adds elements to the clusters in a greedy manner. Several methods have been explored to use k-means approaches without a predefined number of clusters, like X-means by Pelleg and Moore [35]. Many of these methods are computational intense (which often means no real-time processing) and do not work well with a small number of elements.

In addition to an unknown number of clusters, the grouping method should take no or little parameters. Because perceiving groups is a very natural behavior, most of the times requires no knowledge of the drawing's context is required (as stated in Section 2.1.2). Humans perceive groups although the do not know what a drawing means. The context of a drawing can result in a different perception of the containing groups. Parameters for the dynamic grouping process would require knowledge of the context. This could be done with a unsupervised or supervised learning approach to get more specialized information and clusters. The goal of this work is to find general groups of strokes without knowing the context of a drawing.

4.3.3   Hierarchical Clustering

A method that fits all requirements is hierarchical clustering ($= HAC$). Hierarchical clustering is a unsupervised learning algorithm, a good overview and description by Manning et al. can be found in [27, Chapter 17, p. 377-401]. According to Manning et al. [27, p. 399], one of the early mentions was by King in 1967 [21]. It was described as a simple, step-wise clustering procedure. This describes the algorithm pretty well. As with many clustering algorithms, hierarchical clustering is often applied in the field of data analysis or document clustering. Nevertheless, it can be used for any kind of

data that has values to indicate the relation, similarity or distance between the input objects.

Hierarchical clustering takes the pairwise element affinity values as input. In case of the usage to produce perceptual groups from elements, the input are the similarity values between the different elements, calculated in the Feature Extraction.

There are two different approaches of hierarchical clustering. The top-down hierarchical clustering (also revered to it as *divisive clustering* [27]) is an approach that starts with one initial cluster, which contains all input elements, and splits the clusters until convergence. The top-down approach uses another clustering algorithm to split that cluster. All divided clusters are then again split until there are only clusters left that contain exactly one element [27]. This can be done in an recursive manner. The second and more commonly used approach is the bottom-up approach, called *hierarchical agglomerative clustering*. On the contrary to the top-down hierarchical clustering, this approach starts with all elements being separate clusters and merges these clusters until only one cluster containing all elements is left. In this thesis, the hierarchical agglomerative clustering is used and described in detail. The top-down approach needs another clustering approach, like for example k-means clustering, as "subroutine" [27]. Since other clustering approaches often do not fit the requirement, the bottom-down approach was chosen for the dynamic grouping process.

HAC Base Algorithm

The hierarchical agglomerative clustering takes a distance matrix as input, which contains the pairwise affinity values for all elements. Figure 4.9 shows an example of such a distance matrix. The base algorithm assigns each object
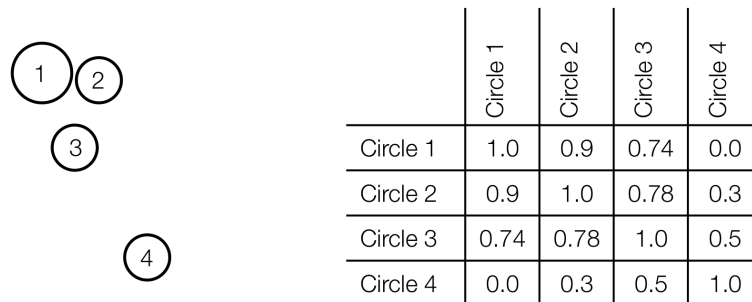
	Circle 1	Circle 2	Circle 3	Circle 4
Circle 1	1.0	0.9	0.74	0.0
Circle 2	0.9	1.0	0.78	0.3
Circle 3	0.74	0.78	1.0	0.5
Circle 4	0.0	0.3	0.5	1.0

Figure 4.9: Example of a 2D distance matrix based on the proximity of the circles on the left side. The values in the matrix on the right side are normalized in a range from 0 to 1. A value of 0.9 means the objects have a proximity of 90% based on the maximum distance.

to a cluster. The result are n clusters, where n is the number of input objects. For the proposed system Suggero, n equals the number of elements in a sketch. It then searches for the two clusters with the shortest distance value (or highest affinity value) in the input distance matrix. Self similarities from the distance matrix are not taken into account. When the two clusters are found, the algorithm merges the two clusters. After merging, there are $n-1$ clusters remaining. All subsequent steps merge the clusters with the best heuristics for belonging together. The algorithm ends when only one cluster containing all n elements is left.

As stated in [27], there are several ways to determine the heuristics on which clusters to merge. If clusters contain only one element, the value from the input matrix is taken as a heuristic. If a cluster consists of more than one element, there are three different heuristics.

Single linkage model: The heuristic for the cluster similarity is the relation (distance) of the two nearest strokes. That way, only the two most similar members of two clusters are used for the comparison of two clusters.

Complete linkage model: The complete linkage model calculates the distance between two clusters by taking the relationship values of the two most dissimilar members in account. By using this heuristic, clusters tend to be fairly specialized and all members of a cluster must have more similar heuristic values than with the single linkage model.

Both heuristics are not optimal to preserve a good group quality, more resulting in groups with high inter-cluster scattering, as pointed out in by Manning et al. [27]:

> "Single-link and complete-link clustering reduce the assessment of cluster quality to a single similarity between a pair of documents: the two most similar documents in single-link clustering and the two most dissimilar documents in complete-link clustering. A measurement based on one pair cannot fully reflect the distribution of documents in a cluster. It is therefore not surprising that both algorithms often produce undesirable clusters."

Group-average agglomerative clustering   Another approach, and the third heuristic, for calculating cluster similarity is to take all the elements of the clusters into account. This approach is called *group-average agglomerative clustering* ($= GAAC$) [27]. All pairwise distance values are used. By averaging these values, a good group quality and coherence can be achieved. The computation is not more intense than for single linkage model or complete linkage model, because those two also require searching through all the cluster elements.

The group-average agglomerative clustering algorithm ends in one cluster containing all input elements. Having one cluster as the result is not the goal of the Dynamic Grouping. The desired result is a list of clusters, ranked by their quality. The first step for obtaining this is to preserve all clusters that are calculated during the grouping process. The resulting clusters have no measure concerning the quality of the clusters. To get a valid indicator for the cluster quality, several additional calculations are performed.

Cluster Quality Measure

The most obvious and probably "natural" way of determining the quality of the cluster created by the GAAC algorithm is to use the average pairwise similarities calculated earlier. All the affinity values of the elements, excluding the self similarities of the cluster, are averaged. This way, an easy and accurate quality value for later ranking is calculated. Averaging the elements' similarities also comes with disadvantages.

First of all, some extracted features are binary, like connectedness. Other features like color or similarity produce same similarity values for groups if all the elements have the same color. For binary features like connectedness it does not matter if a group has two, three or more elements, the group quality is always the same. Secondly, a quality measure should take the group size into account. Thirdly, for features like shape similarity, the group quality decreases with increasing number of elements being added to a group. Hand-drawn shapes by users are never 100% identical. This leads to a static decline in group quality for this kind of features. To overcome these issues, the average affinity values are combined with a penalty function for the group size. This penalty function serves as a tie breaker for clusters with the same average values values. Additionally, it should rank groups with fewer elements higher than larger groups. The parameter γ was determined empirically to be 0.9. This way, groups with fewer elements are always ranked higher than larger groups. This behavior is subject of the later preliminary study.

By preserving all discovered groups and combining them with a quality measurement, all the requirements for the dynamic grouping algorithm are met. The algorithm works well with the input provided by the prior feature extraction (*Input compatibility*) without knowing the number of occurring clusters in advance or setting any fixed parameters (*Non-parametric*). The algorithm produces a hierarchical list of groups, with elements assigned to multiple groups, depending on their similarity (*Fuzziness*). Lastly, the groups can be ordered by their *quality measure* and used for further processing.

Post-processing of Groups

The groups from the specialized feature groups (see Section 4.3.1) are processed after the Dynamic Grouping, because some groups may be redundant.

Since all features are processed separately with the *HAC* algorithm, it might happen, that the same groups are found with different features. An example would be two elements that have the same color and the same thickness would be grouped for both features. In order to not present groups multiple times, duplicates are removed. If a collection of duplicates is found, the group with the highest confidence value is considered for later presentation while the other duplicates are removed.

4.4   Performance

Since sketches change with every new input element or modification, the process of finding perceptual groups in Suggero has to occur in real-time. In order to increase performance of Suggero, several steps have been performed. The processing of the input element was moved from the UI thread of the sketching application to the background. This way, even if Suggero needs several hundred milliseconds to calculate the perceptual groups, the interface does not get unresponsive. Since users have to change tools between drawing and selecting, the calculation can take several hundred milliseconds. If the calculation takes longer, Suggero shows a waiting wheel instead of the suggestions. This can occur for very large sketches and indicates that there still is room for improvement in terms of computation speed.

For the calculation of global proximity, it would have been possible to compare all points of an element. This would have led to a large number of distance calculation. In order to avoid this, only 10 stroke points are compared (see Chapter 4.2.1). For the feature connectedness, only elements with overlapping bounding circles are compared. If the bounding circles don't overlap, the elements cannot have an intersection.

Since the extraction of the features used in Suggero do not depend on each other, it was possible to parallelize them. The sketching application is built on top of the *Windows Presentation Foundation* framework (using .Net 4.0). These frameworks offer the possibility to parallelize tasks using different threads, which was done for Suggero. All calculated properties of the input elements (bounding circle, Fourier descriptors) are preserved and only updated if the input element changes (e.g. gets translated or rotated), which also increased performance.

Chapter 5

Experiment 1: Observation

To gain knowledge about expectations on a perceptual grouping tool and to evaluate our novel perceptual grouping tool called Suggero, two user studies were conducted. Chapter 3 described the design and concepts of Suggero, while Chapter 4 provided details on the used algorithms and the implementation. Both experiments were conducted as controlled laboratory-based experiments to reduce external factors potentially influencing the results. This chapter presents the first experiment, the exploratory study. The experimental design is explained in the first part of this chapter, while the second part covers the results.

In a preliminary study, a small number of participants were drawing sketches using the sketching application (described in Section 3.5.4) without the Suggero technique, and then select drawing elements manually to provide training data for our algorithm.

One goal of this first experiment was to collect a test corpus of drawings and perceptual groups created by humans. Since Suggero uses shape, color, thickness and position as features, the sketches were used to get realistic data for adjusting the weights and parameters. Additionally, a test set for later improvements of the system was collected.

5.1 Participants

10 participants (4 female, 6 male) aged between 20 and 39 years ($M = 26.7$ years, $SD = 5.75$ years) were recruited from a local university. All had experience either with pen-based or touch-based devices, and two of them also reported having experience with interactive whiteboards.

5.2 Apparatus

This study was conducted in a quiet room equipped with a 70" interactive whiteboard. A Hitachi CP-AW251N ultra short throw projector with

a resolution of 1280×800 pixels was used for the projection (see Figure 5.1). Input was provided with an Anoto digital pen (ADP-301). To perform the experiment, participants used a sketching application on the interactive whiteboard. The sketching application is described in Section 3.5.4. The Harpoon selection tool (described in Section 3.5.3) was used for all selections.
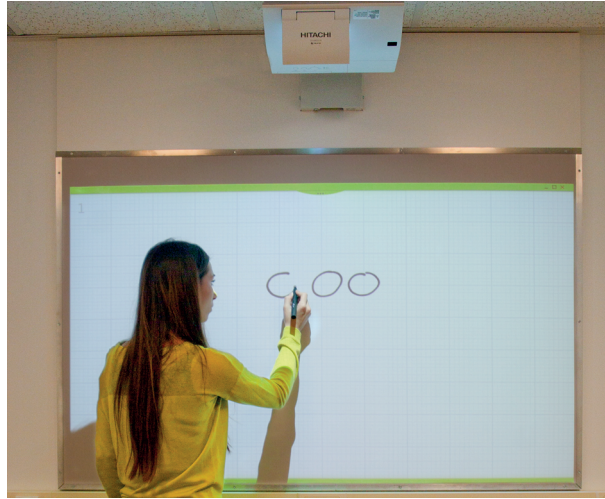
Figure 5.1: The setup for the experiment, a 70" interactive whiteboard operate with Anoto digital pens and a Hitachi CP-AW251N ultra short throw projector above the whiteboard used for projection.

5.3   Experimental Design

A repeated-measures design was used for the experiment. The independent variable was Drawing with four levels (*Drawings A*, *Drawing B*, *Drawing C*, *Drawing D*). Four blocks were completed, each consisting of one drawing. In each drawing, five different strokes were chosen by the system to be *candidates* for group selections. A *candidate* was a highlighted element for which participants had to find perceptually salient groups. The candidate strokes were randomly chosen for every participants. Participants were asked to select the four visually most salient groups of drawing elements for each candidate, in decreasing order of visual obviousness. This resulted in 10 participants $\times$ 4 drawings $\times$ 5 candidates $\times$ 4 groups = 40 user generated drawings and 800 selected groups.

No counterbalancing was needed, since only one selection technique was used and the order of blocks did not influence users behavior in terms of group selection. Each participant performed the four blocks in the same order (*Drawing A, Drawing B, Drawing C, Drawing D*).

5.4   Tasks

Four drawings showing realistic scenes were drawn by the participants. The scenes were provided as templates in form of simple vector drawings on paper. The four chosen scenes are a trade-off between covering a large area of possible sketching types and keeping the study within 60 minutes to avoid fatigue. The scenes were

- a landscape scene (*Drawing A*, Fig. 5.2 a),
- a simple map with direction instructions (*Drawing B*, Fig. 5.2 b),
- a mind-map (*Drawing C*, Fig. 5.2 c),
- and an interface of a fictive application (*Drawing D*, Fig. 5.2 d).

Participants were instructed to follow the template regarding positioning and relative size. The drawings were provided as black and white images and participants and were instructed to use at least four different colors and two different stroke thicknesses to get more variation in the resulting sketches.
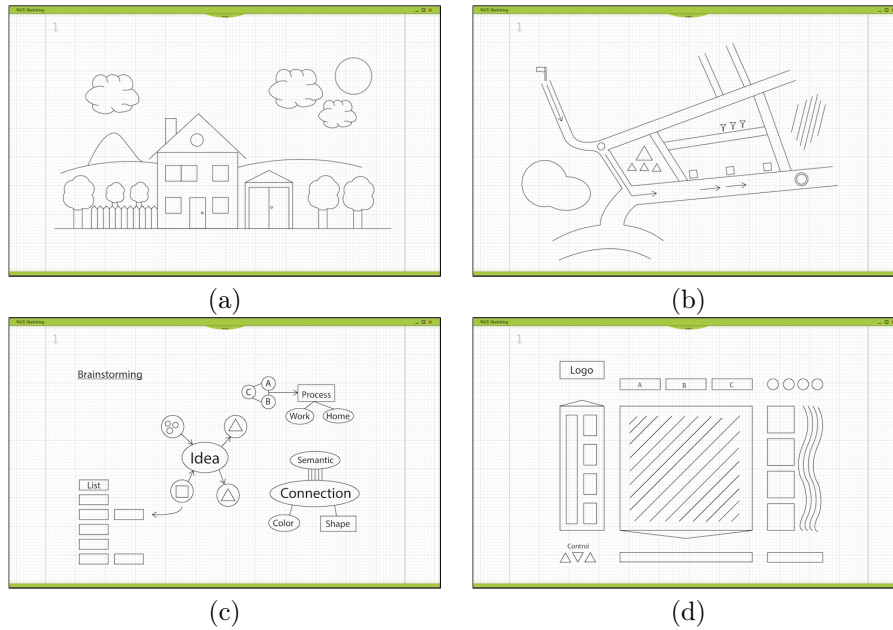
Figure 5.2: The four templates. (a) is the landscape scene; (b) is a simple map with direction instructions; (c) a mind-map; (d) a simple interface concept of a non-existing application.

After completion of each sketch, a button vertically centered on the right side of the screen was pressed to proceed with the experiment. Next, the system successively highlighted a random stroke in the drawing, the current *candidate*. Participants were instructed to select the four most salient and

visually obvious groups containing the candidate, starting with the most salient group. After each selection, again a vertically centered button at the right side of the screen was pressed to indicate completion of the current selection. The Harpoon selection tool was used for all selections. The selection procedure was performed for each of the five candidates (with each time four group selections). After the last selection in the block was completed, participants went on to the next block. Each block was performed in approximately 10 minutes.

5.5   Procedure

In a 10-minute training period, the whiteboard and sketching application was explained and participants drew a training sketch. Afterwards, partipants went on with the experimental tasks. The four drawings were drawn one after each other. In each trial, a candidate drawing element was randomly chosen and the participant was asked to identify four groups of elements that corresponded to the candidate, in descending order of relevance. Selections were performed with the Harpoon selection tool, described in Section 3.5.3. Each session lasted approximately 60 minutes, and each participant performed a total of 80 trials (4 drawings $\times$ 5 candidates $\times$ 4 groups).

5.6   Data Collection and Analysis

A full-factorial repeated-measures ANOVA ($\alpha = 0.5$) was performed on the factors Drawing, Candidate, and Group. The analyzed measures were *group size*, as being the number of selected elements in a group, and *selection time*, defined as the time between tapping the start and end button. Data was collected using an extensive log file. Time was measured and analyzed in milliseconds but is presented in seconds for better understanding. The Greenhouse-Geisser correction was used if the assumption of sphericity was violated. Post-hoc analysis on the main effects were performed, including paired-samples t-test using Bonferroni adjustments.

5.7   Qualitative Analysis

The groups were used for tuning the parameters of Suggero. With information about candidate and selected groups, parameters and weights of the algorithms were revised. The parameter tuning was done manually and did not cover a complete tuning of all parameters. Since the weights and parameters of Suggero are continuous values, there is an infinite number of possible combinations. The collected test corpus can be used for automatic parameter tuning, which is out of the scope of this thesis. Tuning the parameter manually gave good results, but there is definitely room for improvement.

5.8 Quantitative Analysis

Two different measures were analyzed, *group size* (number of elements in selection) and *selection time* (in seconds).

5.8.1 Group Size

Statistical analysis showed main effects for *Drawing* $F_{3,27} = 9.873, p < .001$ and *Group* $F_{3,27}, p < .001$. Pairwise comparisons of the factor *Drawing* showed that the group size for *Drawing B* is significantly higher ($p < .01$) than for *Drawing C* (the mind-map). Figure 5.3 shows that the four drawings were different in terms of group size. There was no pairwise significance in group size between the other drawings. The not significant differences indicate that although the group size varies between the drawings, some perceptual groupings are the same.
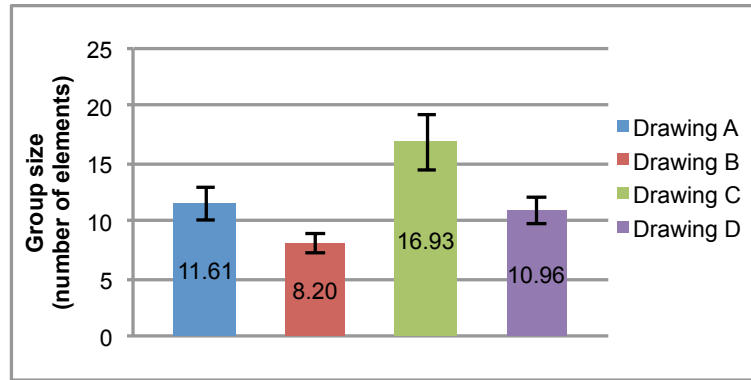
Figure 5.3: Group size for each *Drawing*.

Although the group size varied between the different drawings, there was a significant main effect for *Group*. Figure 5.4 shows the group size for the different groups with standard error. The group size increase significantly between *Group 1* and *Group 2* ($+67.27\%, p < .01$) and between *Group 2* and *Group 3* ($+36.15\%, p < .01$). Group size also increase between *Group 3* and *Group 4*, although it did not reach significance ($p = .437$). Additionally, there was a pairwise significance for group size between *Group 1* and *Group 3* ($p < .001$) and between *Group 1* and *Group 4* ($p < .05$). No significant difference was found between the other drawings. Since participants selected more obvious groups first, this results indicate that larger groups are less visually obvious than groups with fewer elements. This insight is reflected in the calculation of the group quality value described in Section 4.3.3. Groups with fewer elements were preferred over larger groups. The penalty function for group size is directly related to number of elements in a group. This means that the penalty value is higher for larger groups and less for groups with
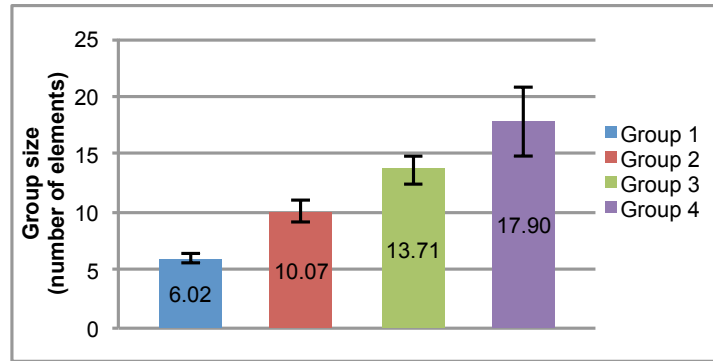
Figure 5.4: Group size by *Group*.

fewer elements. In addition to the main effects, results showed an interaction between *Drawing* and *Group* $F_{2.623,23.609} = 4.436, p < .05$. This effect is suspected to be because of the the variety in the group size of the drawings. Still, the number of elements per groups increased with the *Group*, which reflects the main effect on *Group*. Figure 5.5 illustrates this interaction.
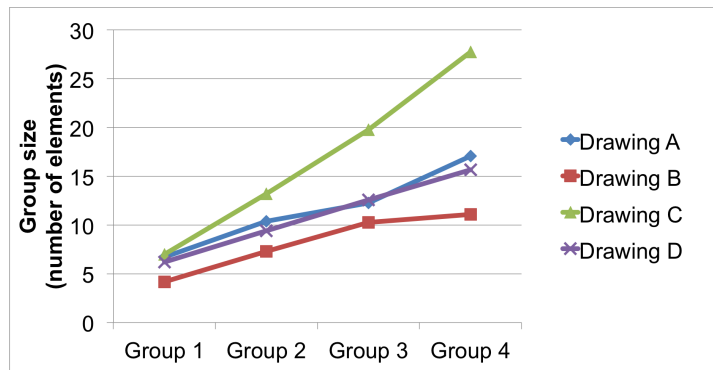
Figure 5.5: Group size for *Drawing* by *Group*.

5.8.2 Selection Completion Time

Results showed a main effect for *Drawing* $F_{3,27} = 10.356, p < .01$ and *Group* $F_{3,27} = 4.694, p < .001$. The selection completion time was very different across the drawings (see Figure 5.6).

The first two selections of the groups took participants nearly the same time. There was a significant increase in time needed after *Group 2*. Participants needed significantly longer for selecting *Group 3* ($46.23\%, p < .05$) and Group 4 ($p < .05$). These results are illustrated in Figure 5.7.

Since it took participants longer to come up with groups after *Group 2*, it
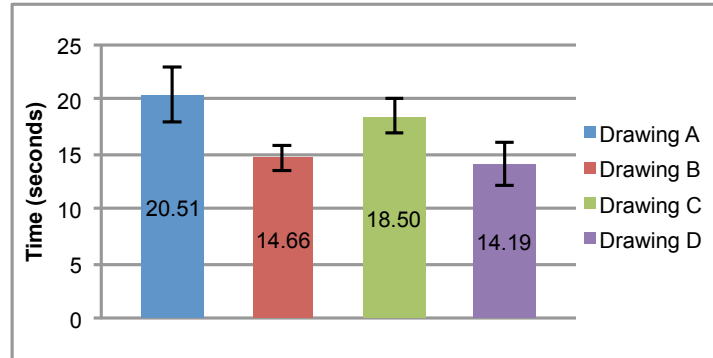
Figure 5.6: Selection time for each Drawing.
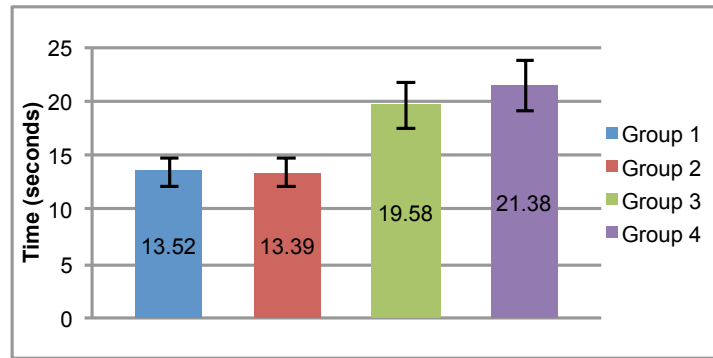
Figure 5.7: Selection time for each Group.

indicates that *Group 3* and *Group 4* were more difficult to find. Igarashi et al. [17] and Suggero both present three suggestions. The significant gap in time between *Group 2* and *Group 3* indicates that showing two *correct* suggestions would be sufficient. By using two instead of three suggestions, space needed for the suggestions as well as clutter on the screen could be decreased. Still, since it can not be ensured that the suggestions provided by the system are correct, the third suggestion may be seen as a *backup suggestion*, in case the first two suggestions are incorrect. This *backup suggestion* may not be the best solution. A good mode switching between showing and hiding the suggestions could provide users with better support for selection tasks.

Chapter 6

Experiment 2: Comparative Study

This chapter present the second experiment conducted to evaluate Suggero. Chapter 3 described the design and Chapter 4 the implementation of Suggero. Chapter 5 described the first user study with details on the experimental design and the results.

While it may seem clear that for the selection of a large number of elements, some assistance can prove useful, there remains an open question about whether providing suggestions for selections will hinder or help. Specifically, the performance gain achieved by reducing a large number of selections to a single tap may be outweighed by the cognitive load required to identify that group among others and to switch tasks between selecting strokes and identifying that group. Our second study was designed to examine this tradeoff, as well as to provide the opportunity to make observations about how people use our technique. This study consisted of two parts: in the first part, we compared Suggero to the Harpoon selection tool; in the second part, we observed participants using Suggero to modify a realistic drawing to get insights about how it is used

Experiments in Current Literature

It was important to find an experimental design which leads to *comparable*, *reproducible* and *unbiased* results. Besides these criterions, several other factors are important for a suitable study design to test a perceptual grouping system. Collecting *quantitative data* to measure performance in terms of *speed* and *accuracy* should be possible. Furthermore, *comparison* of the data between the perceptual grouping system and another system or selection technique must be possible, in this study Suggero and Harpoon. For Suggero, this means that the tasks performed in an experiment should cover its strengths and weaknesses. To explore and develop such an experimental design, experiments in current literature were reviewed to find an approach that

meets all criterions. Current literature on perceptual grouping provides no information about conducted user studies with the goal of collecting quantitative data. Thórisson [40] and Igarashi et. al [18] presented experimental results, while Wuersch and Egenhofer [45] concentrated on verifying automatically labeled regions in user sketches. Interesting studies have been conducted in the field of selection techniques. Mizobuchi and Yasumura [28] tested their selection technique by letting users select predefined objects in a grid consisting of rectangles. By selecting a predefined group of objects, collecting quantitative data on accuracy and performance was possible. Dehmenshki and Stuerzlinger [9] also used this approach to test their perceptual-based object group selection technique. They added two additional complexities by varying the spacing between the objects and by rotating the whole grid in different angles. Grossman et al. [13] tested two different kinds of drawings in their study. The first one was also a grid consisting of primitive objects (crosses), which was later adopted and extended by Leitner and Haller [24]. The second type of drawings tested by Grossmann et al. were hand-drawn scenes. They tested their interaction technique "Handle flags", which provides users with suggestions from a perceptual grouping algorithm. They focussed strongly on the interaction part, so their "automatically found" groups were manually pre-defined. Users were provided with the finished sketches of realistic scenes, thus, defining the suggestions for a manually selected element in advance was possible. Since these studies focused on performance and comparison of different selection techniques, their method can be partially adopted.

6.1   Complexity

In order to be able to compare Suggero and Harpoon under different levels of complexity, the different factors of complexity for both tools were explored. Perceptual grouping systems are created to support users in performing selections of perceptually related elements. Using different complexities regarding target selections as independent variables is a common approach in current literature for conducting comparative studies of selection techniques [9, 13, 24, 28]. Complexity for a perceptual grouping system like Suggero partly correlates with the complexity for a regular selection tool. Besides *selection complexity*, *visual complexity* is a good measure of complexity for perceptual grouping systems. To control the complexity of the target selection, current literature uses abstract representations of drawings [9, 13, 24, 28]. These abstract representations consist of abstract objects, most of the time primitive shapes like circles or crosses. To get different levels of complexity, the abstract representations varied in terms of cohesiveness of the objects or rotation. In order to find tasks with different levels of complexity, the term "complexity" was evaluated.

In the context of perceptual grouping and this experiment, the term complexity is a mix of different factors. On the one hand, there is *visual complexity*, with itself being a compound of the complexity of single shapes and the content of the sketch. On the other hand, *selection complexity* indicates the difficulty to perform a particular selection.

6.1.1 Visual Complexity

As with human perception of groups, there exists no exact measure or predictive model that explains how humans perceive complexity. Early research on complexity calculation goes back to the work of Arnoult and Attneave in 1956 [3] and Attneave in 1957 [2]. After conducting a study with 168 participants ranking random shapes for their complexity (without prior explanation of the term "complexity"), they found several indicators for the perceived complexity of shapes. Besides curvedness and the perimeter-to-area-ratio, the most important feature they found was number of turns of a shape [2]. Palmer points out (in his outstanding book "Vision Science: Photons to Phenomenology") that, although number of turns can be a good indicator, this must not necessarily be the case [32, page 399]. There exists an infinite number of objects with the same number of sides, with different (subjective) ratings of complexity. An example is shown in Figure 6.1. Palmer also uses "Figural Goodness" to explain subjective perception of complexity [32, page 398].

> "Figural goodness is the aspect of perceptual experience that is perhaps best described as a composite of (at least) the simplicity, order, and regularity of an object."

This indicates that complexity of a shape, an image or a drawing is a combination of multiple features. Sets of objects are perceived most complex if no object equals another. As Oliva et al. state in [30],

> "... visual complexity is principally represented by the perceptual dimensions of quantity of objects, clutter, openness, symmetry, organization, and variety of colors."
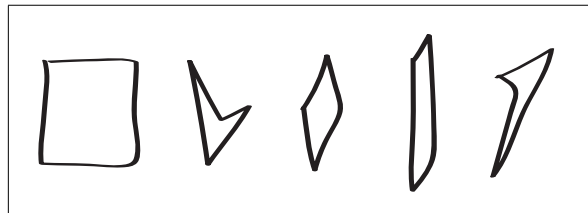
Figure 6.1: Although all of the objects have 4 turns (sides), the subjective complexity is different.

An illustration of different complexities based on these indicators is shown in Figure 6.2. If a drawing has a high degree of complexity, it is more difficult for humans to perceive visual structures and groups. For testing the system with different controllable levels of complexity, several indicators for visual complexity had to be combined.

(a)   (b)
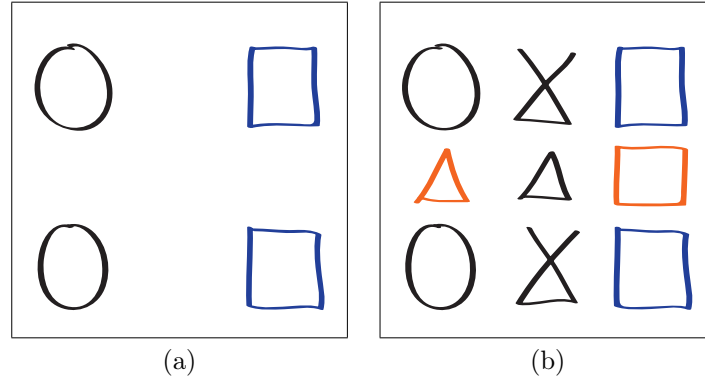
Figure 6.2: Differently complex sketches. (a) is perceived as less complex, because of the lower quantity of different shapes (2), colors (2) and subgroups (2). (b) is more complex with higher number of different shapes (4), colors (3), shape-color combinations, and many possible subgroups.

Summarizing, visual complexity is influenced by a variety of factors, some of them are based on the complexity of the shapes themselves, like the numbers of turns of a shape or its transformation invariance. Additionally, visual complexity is guided by factors that concern the content of a sketch, which are

- variety of shapes,
- variety of colors,
- overall number of shapes
- number of subsets
- and cohesiveness.

6.1.2 Selection Complexity

Complexity of selections in a digital drawing or sketching environment has been subject of extensive research. Like many others, Mizobuchi and Yasumura [28] used Fitts' law [12, 26] and the Hick-Hyman law [15, 16] as a foundation to explain complexity of a target selection and to predict the required time for a selection. Accot and Zhai [1] additionally introduced the Steering law to predict selection time. Applying these laws for receiving different levels of complexity often results in usage of patterns with different cohesiveness of objects. Dependent on the selection technique, the mentioned

laws influence the selection complexity and time. The Steering law for example does not apply to tapping selections in a two-dimensional environment (like a drawing). There is no need for the user to exactly follow object paths and boundaries to select objects. On the other hand, methods like crossing or circling depend directly on the Steering law to perform a correct target selection. Summarizing, the most influential factor for selection complexity are

- the distance of targets (due to Fitts' law),
- number of distractors (due to Hick-Hyman law and Steering law),
- cohesiveness of the content (due to Steering law),
- number of objects in the target selection (due to Hick-Hyman law and Fitts' law)
- and the size of the elements in the target selection (due to Fitts' law).

An illustration of different levels of selection complexity can be found in Figure 6.3.

(a)   (b)
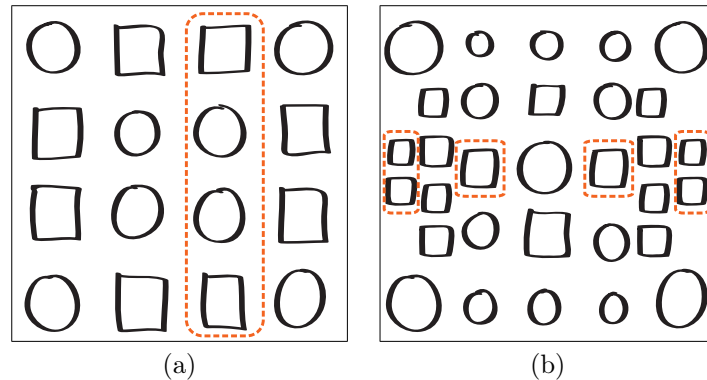
Figure 6.3: Differently complex selections. In (a), there are few distractors for the selection (objects within dashed area), the spacing between the objects is large, the size of the target objects is large and there are only few objects to be selected. In (b), the target selection is farther apart, more distractors are present and the target objects are more and smaller objects.

6.2 Participants and Apparatus

18 paid participants were recruited (10 males, 8 females) from a local university. Participants ranged from 22 to 40 years in age ($M = 29$ years, $SD = 8.16$ years). All participants (2 left-handed, based on self-reports) controlled the stylus with their dominant hand. Five participants reported having experience with interactive drawing applications; eleven participants reported having no experience in working with interactive whiteboards. The

apparatus was the same as in the first experiment (described in Section 5.2). Additionally, the drawings and target selections were shown on a projection on the right side of the participants in the same scale as the later sketch.

6.3   Experimental Design

The second experiment consisted of two parts, with the first part being a comparative study and the second part being an observation.

In the first part of the experiment there were two independent variables, namely *Technique (Suggero, Harpoon)* and *Complexity (Simple, Challenging, Arbitrary)*. A 2×3 design resulted in six conditions. Presentation order of the *Complexity* was counterbalanced using a 3×3 latin square, resulting in each order completed by six of the 18 participants. To counterbalance *Technique*, three of the six participants started with *Suggero* for each *Complexity* and three started with *Harpoon*. Each block lasted approximately 20 minutes, This part was split by three blocks, each having a different level of complexity. In each block, participants had to redraw a template and make 20 selection with each of the both techniques. This resulted in a total of

$$
\begin{array}{rl}
 & 20 \quad \text{Suggero selections} \\
+ & 20 \quad \text{Harpoon selections} \\
\hline
 & 40 \quad \text{target selections performed per block.}
\end{array}
$$

6.3.1   Factors

The factors for the user study were *Technique* and *Complexity*.

Technique

The proposed technique Suggero, a selection technique for perceptual groups, was compared with Harpoon, a manual selection technique. The Harpoon selection method [24] is a speed-dependent crossing technique that supports users in performing complex selections (described in Section 3.5.3). The initial selections used to determine the suggestions provided by Suggero were also performed using Harpoon selection.

Complexity

A fundamental aspect of the tradeoff between improving performance by suggesting completions and interfering with the cognition required to perform the initial selection is the idea of complexity. Specifically, it was suspected that there would be a "sweet spot" of complexity in which the selection was sufficiently complicated that manual selection was tedious, but simple

enough that Suggero would still be capable of providing reasonable suggestions. Thus, in addition to varying whether or not suggestions were provided, this level of complexity was adjusted, too. Using different complexities for target selections when evaluating selection techniques is a common approach, as stated earlier in Section 6. Two important factors for complexity are the visual complexity (see Figure 6.4, *top*) of the content and selection complexity (see Figure 6.4, *bottom*) of the target selection. The factors to control visual complexity and selection complexity are explained earlier in Section 6.1. Sketches are perceived to be the most complex if no object equals another and no visually salient subsets are present. Visual complexity and selection complexity were combined to create three levels of the complexity condition: *Simple*, *Challenging*, and *Arbitrary* (see Figure 6.5).
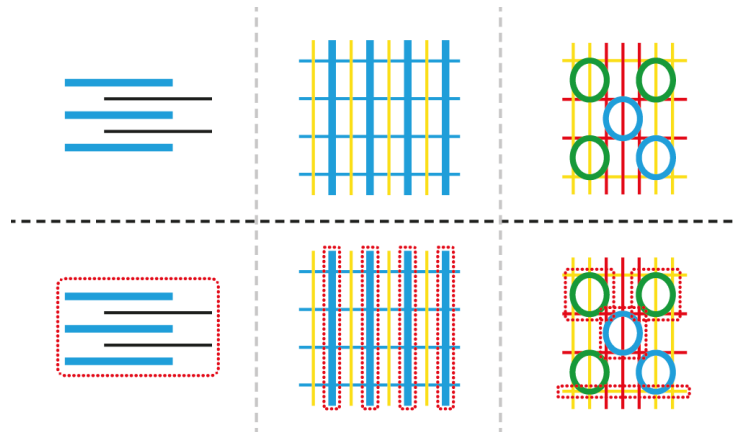
Figure 6.4: *Simple* (*left*), *Challenging* (*middle*) and *Arbitrary* (*right*). Vectors graphics (*top*) and target selections (*bottom*) that were provided to participants. Target selections included all elements completely within a dotted area.

Simple sketches had low visual complexity and low selection complexity for both techniques. *Challenging* sketches were slightly more visually complex and had low selection complexity for Suggero, but a high selection complexity for manual selection. This *Challenging* condition represents circumstances when many elements have been drawn and are perceptually related, but might interfere with manual selection. *Arbitrary* sketches had both high visual complexity and high selection complexity for both selection techniques (see Figure 6.4, *right*). These sketches were generated by arbitrarily choosing elements to render, thus minimizing perceptual relationships. The challenging sketches were expected to be the "sweet spot" in which Suggero would outperform manual selection, but that manual selection would outperform Suggero for simple selections. The benefit of Suggero was expected would no longer hold for the arbitrary condition. Abstract representation of realistic
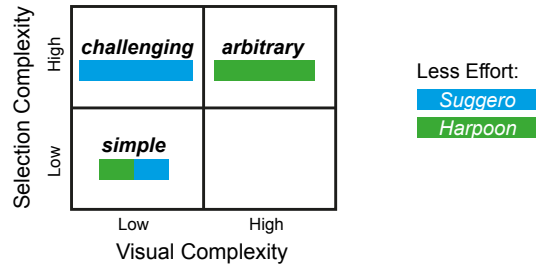
Figure 6.5: The different conditions of complexity based on their properties.

scenes, abstract drawings, were used to control the different factors of visual and selection complexity. Participants were asked to draw the abstract drawings themselves to avoid biasing the study in favor of Suggero and to be able to analyze the performance of Suggero in differently drawn sketches.

6.4 Task Design

Each of these three complexities (*Simple*, *Challenging*, *Arbitrary*) was represented in an abstract drawing and its target selections. The corresponding abstract drawings can be found in Figure 6.6.

In addition to represent the different complexities in abstract drawings, the target selections were matched to fit the particular complexity. For each of the abstract drawings, 30 to 40 target selections matching the complexity of the factor were manually created. From this set of selections, 20 target selections per drawing were randomly chosen before the experiment and used as target selections. Selections were indicated in the abstract drawings with dashed rectangles. Participants were instructed to select all elements within the area of such a rectangle. Samples of the used target selections are shown in Figure 6.7. All target selections can be found in Appendix A.

6.5 Procedure

Participants were briefly introduced to the experimental setup and the purpose of the experiment, followed by a 15 minute training session. During the training, participants were guided through the process of creating a sketch using a practice drawing and performed a minimum of 5 selections with both techniques. For each complexity, participants were asked to select elements on a sketch provided by the experimenter. Participants began each trial by tapping the start button, the target selection was then shown at the participant's right, and once they had performed the selection, would end the trial by tapping the end button. Participants were instructed to perform selections as quickly and accurately as possible. With Suggero, participants were

Simple

Challenging

Arbitrary
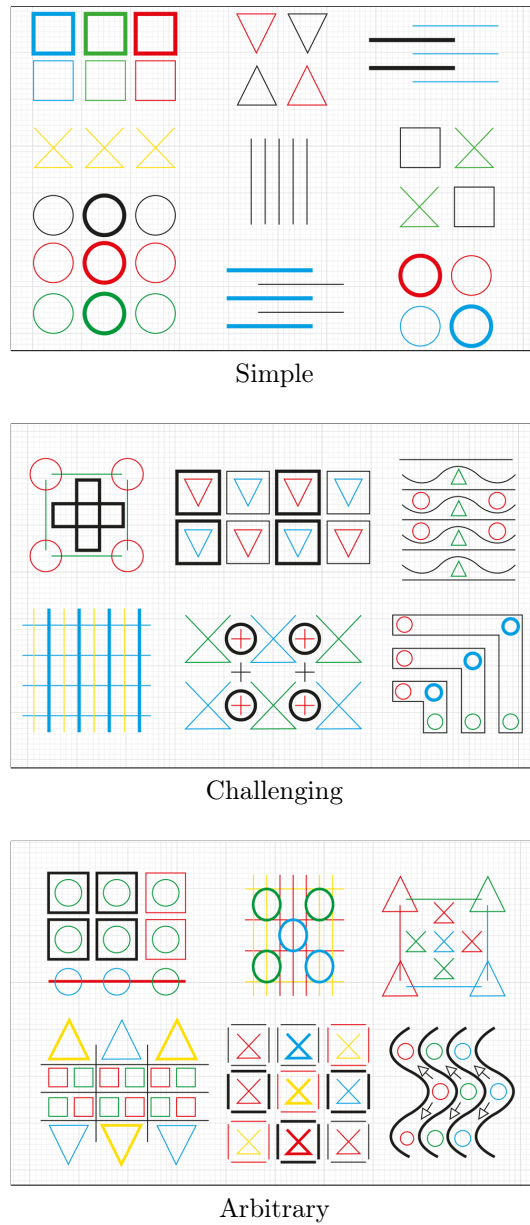
Figure 6.6: The used abstract drawings for the different conditions of *Complexity*.

additionally asked to use the smallest number of manually selected elements, in order to encourage them to use the provided suggestions.

After completing all trials for each technique, participants filled out a questionnaire. The order of complexities was counterbalanced using a Latin square. Each complexity corresponded to an abstract drawing and target
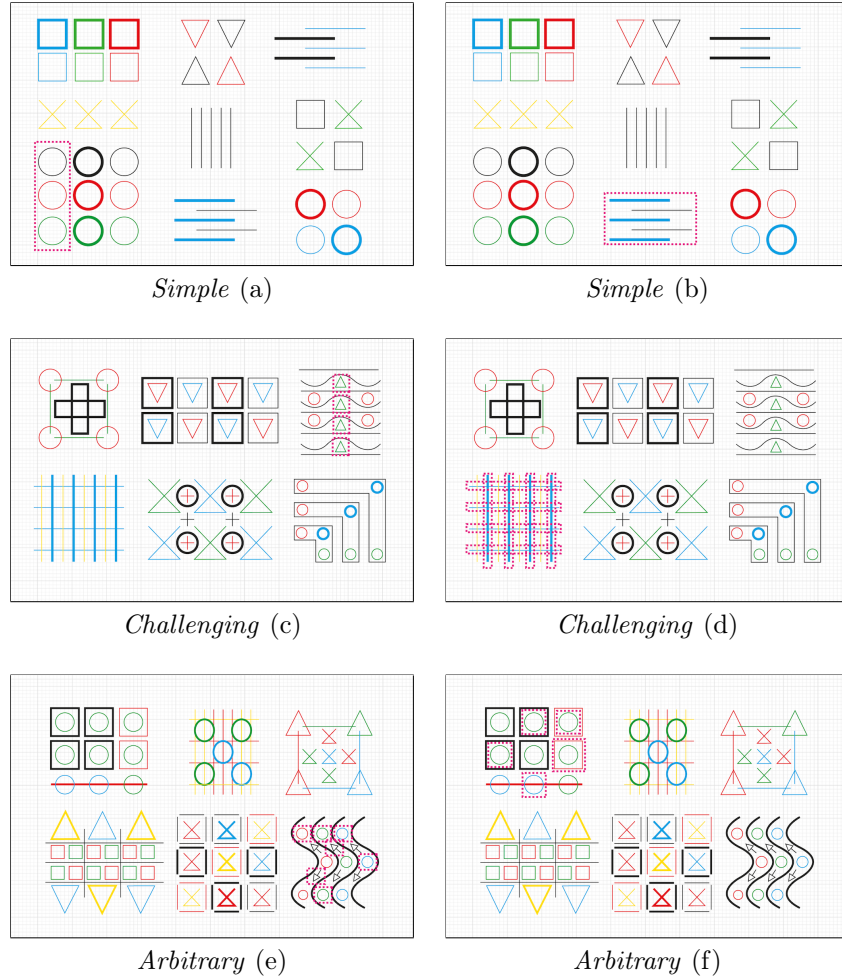
Figure 6.7: Examples of target selection for the three different complexities. *Simple* is shown in (a) and (b), *Challenging* in (c) and (d) and *Arbitrary* in (e) and (f).

selections specific to that drawing. Selections were determined in advance (same for each technique and all participants). Each participant performed a total of 120 trials (2 techniques $\times$ 3 complexities $\times$ 20 selections).

In the next phase of the study, an incomplete sketch (with 30% of the strokes manually removed from the initial drawing) was provided and participants were asked to continue a drawing (2 participants' drawings from the preliminary study) in whatever way they desired for 5 minutes. Participants were then asked to perform selections with Suggero and to interact with the selected elements (moving, rotating) in a 5 minutes speak-out-loud session. The entire session lasted approximately 90 minutes.

6.6 Hypotheses

The study was conducted in respect to the following hypotheses:

Hypothesis 1: *Suggero returns the intended results with one or two manually selected strokes.* Users have the possibility to select multiple elements manually. After selecting the elements, the suggestions are presented. The more elements were selected, the more information the system gets about which group is intended. Suggero is expected to find the desired results by selecting only one or two strokes manually.

Hypothesis 2: *Suggero outperforms Harpoon for complex group selections, whereas Harpoon will perform better for small and simple groups.* Depending on the selection method, users may not need suggestions for simple groups, because the intended selection can be performed with Harpoon easily. Due to this, users will prefer manual selection for small (small area and small number of strokes) and simple groups. For selecting more complex groups that take up a larger area, using a suggestion from Suggero will have advantages for users.

Hypothesis 3: *Suggero outperforms Harpoon for complex selections with obvious perceptual groupings. Harpoon performs better for target selections with no visually salient groups.* Suggero is expected to outperform the Harpoon selection tool when the target selection is a visually salient structures. If target selections are arbitrary (perceptually random), Harpoon will perform better because Suggero can not take advantage of its feature extraction and grouping algorithms. This hypothesis differs from *Hypotheses 2* by considering highly complex groups whereas *Hypothesis 2* is considering simple groups with salient structures.

6.7 Data Collection and Analysis

Stylus movement on the interactive whiteboard and every action in the application was logged and all sessions were audio and video recorded. A 3 (*Complexity*) × 2 (*Technique*) repeated measures ANOVA ($\alpha = .05$) was performed, on four following dependent measures: task completion time, movement time, interaction count, and movement distance. The Greenhouse-Geisser correction was used when Mauchly's test of sphericity was violated (influencing df, F, and p values). Bonferroni adjustments were used for post-hoc analyses.

All sessions were video and audio recorded. Besides that, the system collected data about the time, drawn and selected strokes, the suggestion provided by the system and their usage. Additionally, all interactions while

performing selections within a task were recorded. Participants filled out questionnaires after each of the conditions of the first part of the experiment, including questions about their impression of the system, their opinions and estimations of their performance. The second part of the experiment was speak-out-loud session. All the participants' drawings were saved by the system to extend the test corpus for future work on improving the algorithms and the overall system. Time was recorded and analyzed in milliseconds but is presented in seconds for a better understanding.

6.8   Results and Discussion

This section presents the results and their discussion from the second experiment conducted. The hypotheses for the experiment were described in Section 6.6. Main subjects for the quantitative analysis were the time and interactions.

6.8.1   Trial Completion Time

Trial completion time was defined as the time between tapping the start and end buttons. There was a main effect of *Complexity* ($F_{2,34} = 168.389, p < .001$). Post-hoc pairwise comparisons revealed that all three complexities were significantly different ($p < .05$). Participants were fastest for *Simple* ($M = 4.57$ s, $SE = 0.34$ s), followed by *Challenging* ($M = 8.75$ s, $SE = 0.59$ s) and *Arbitrary* ($M = 19.12$ s, $SE = 1.05$ s). Results showed a main effect of *Technique* ($F_{1,17} = 88.266, p < .001$) with Harpoon ($M = 7.85$ s, $SE = 0.56$ s) being faster than Suggero ($M = 13.97$ s, $SE = 0.7$ s).

There was also an interaction between *Complexity* and *Technique* ($F_{2.34} = 29.518, p < .001$). Post-hoc tests revealed that for each level of *Complexity*, all pairwise differences between techniques were significant (p < .05); however the difference between the two techniques was larger for *Arbitrary* (Suggero: $M = 25.43$ s, $SE = 1.60$ s; Harpoon: $M = 12.81$ s, $SE = 1.03$ s) than for *Simple* (Suggero: $M = 6.18$ s, $SE = 0.39$ s; Harpoon: $M = 3.57$ s, $SE = 0.33$ s) and *Challenging* (Suggero: $M = 10.31$ s, $SE = 0.73$ s; Harpoon: M = 7.19 s, $SE = 0.55$ s).

It was suspected that Harpoon was faster due to the experimenter's instruction to minimize the number of manually selected strokes when using Suggero, as participants were observed spending time determining a strategy to perform a target selection. This led to participants being faster for *Simple* and *Challenging* sketches with the Harpoon technique (which omitted these instructions). Thus, in order to better understand the components of action required to perform selections, we broke down our dependent measure into: movement time, movement distance, interaction count.

6.8.2 Movement Time

Movement time was calculated as the total amount of time per trial that the stylus was touching the surface. Results showed a main effect of *Complexity* ($F_{2,34} = 10.674, p < .001$) with increasing time between *Simple* ($M = 0.63$ s, $SE = 0.058$ s), *Challenging* ($M = 0.74$ s, $SE = 0.10$ s) and *Arbitrary* ($M = 1.22$ s, $SE = 0.16$ s), and all pairwise differences were significant ($p < .05$).

Additionally, a main effect for *Technique* was found ($F_{1,17} = .7.651, p < .05$) with Suggero ($M = 0.69$ s, $SE = 0.06$ s) requiring significantly less movement time than Harpoon ($M = 1.04$ s, $SE = 0.14$ s). Pairwise post-hoc tests showed that Participants spent less time with Suggero than with Harpoon for the *Simple* ($p < .001$) and *Challenging* ($p < .05$) condition. For *Arbitrary*, the difference was not significant ($p = .280$) (see Figure 6.8).
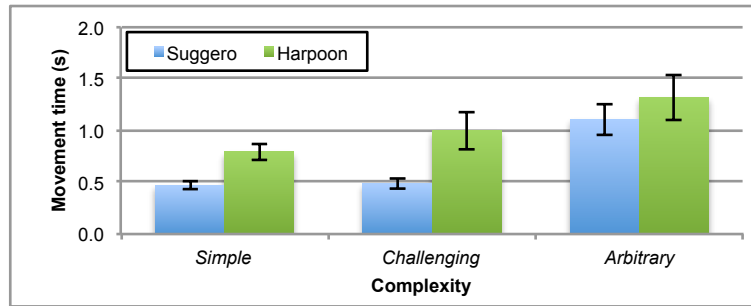
Figure 6.8: Movement time for each *Complexity*.

Although Suggero had longer trial completion time, a closer look revealed that Suggero required less movement time for *Simple* and *Challenging*. Movement time is an important factor for supporting users and to avoid fatigue.

6.8.3 Movement Distance

Movement distance was defined as the distance participants moved the stylus on the interactive whiteboard in pixels. There was a main effect of *Complexity* ($F_{2,34} = 199.382, p < .001$). Post-hoc pairwise comparisons revealed that the difference between *Simple* ($M = 153.05$ px, $SE = 9.82$ px) and *Challenging* ($M = 94.53$ px, $SE = 13.00$ px) was significant ($p < .01$) as well as the difference between *Challenging* and *Arbitrary* ($M = 134.16$ px, $SE = 13.66$ px, $p < .05$). The difference between *Simple* and *Arbitrary* was not significant ($p = .579$).

Results showed a main effect of *Technique* ($F_{1,17} = 56.319, p < .001$), with participants moving the stylus significantly less with Suggero ($M = 72.51$ px, $SE = 5.71$ px) than with Harpoon ($M = 180.65$ px, $SE = 15.55$ px). There was also an interaction between *Complexity* and *Technique*

$(F_{1.391,23.642} = 18.881, p < .001)$. Pairwise post-hoc tests revealed that Suggero required less movement for all three *Complexity* conditions ($p < .05$, see Figure 6.9).
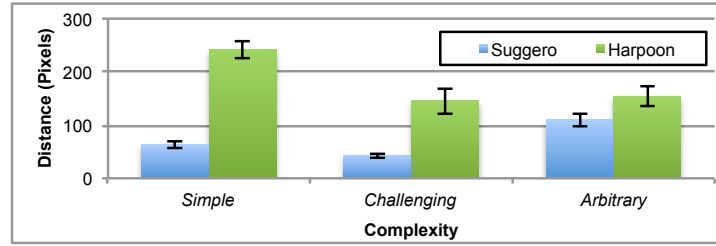
Figure 6.9: Movement distance by Complexity and Technique measures in pixels.

As with the movement time, Suggero required significantly less movement of the pen in terms of distance. Since selections of greater number of elements normally require more movement, avoiding it is important, especially for distant elements like in the *Challenging* condition.

6.8.4   Interaction Count

Interaction count was defined as the number of times participants performed a stroke (touch, optional move, and lift of the pen). Interactions with the suggestions provided by Suggero were also included in this interaction count.

Target group size   The group size of the target selection in the different conditions of *Complexity* is an important information before looking at the interactions. The target groups size is an important factor, since smaller groups most of the time involved less interaction to perform the selection. Analysis on the group size showed a main effect for Complexity ($F_{2,24} = 815.600$, $p < .001$). Group size increase highly significant over the different levels of Complexity (all $p < .001$). For *Simple*, the mean of the group size was 3.14 ($SD = 1.15$), for *Challenging* $M = 4.53$ ($SD = 2.03$) and for *Arbitrary* $M = 5.33$ ($SD = 1.83$).

For interaction count, results showed a main effect of *Complexity* ($F_{2,34} = 155.856, p < .001$) with increasing interactions per *Complexity* (*Simple*: $M = 1.90, SE = 0.08$; *Challenging*: $M = 4.136, SE = 0.21$; *Arbitrary*: $M = 6.31, SE = 0.26$), which were all pairwise significantly different ($p < .05$). Results showed a main effect of *Technique* ($F_{1,17} = 15.075, p < .001$) with Suggero ($M = 4.43, SE = 0.15$) needing more interactions than Harpoon ($M = 3.80, SE = 0.17$). There was also an interaction between *Complexity* and *Technique* ($F_{1.143,19.438} = 12.821, p < .01$). Pairwise post hoc

tests revealed that Harpoon needed significantly fewer interactions for *Simple* ($p < .001$) and *Arbitrary* ($p < .05$). For *Challenging*, Suggero needed significantly less interactions ($p < .05$).

This interaction can be seen in Figure 6.10, which also indicates through shading when Suggero interactions were with suggestions. Harpoon required fewer interactions for *Simple* and *Arbitrary*, while Suggero requires less interaction for *Challenging*. Taking a deeper look on the kind of interaction reveals that a large part of interactions for *Simple* and *Challenging* are interaction with suggestions, which are basically just tapping actions. These interactions require neither much time nor effort. The design of Suggero targets exactly these kinds of selections with reducing the interaction effort to perform selections.
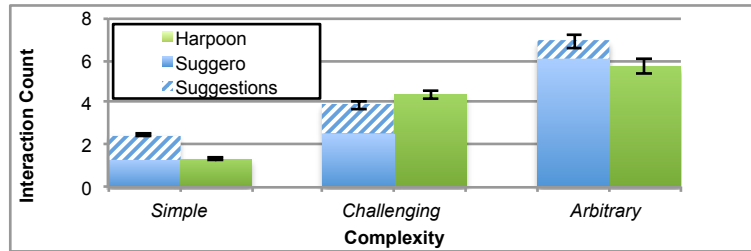
Figure 6.10: Interaction count by Complexity and Technique. Shaded areas indicate interactions with suggestions.

6.8.5   Suggero Usage

Besides the repeated measures analysis, the usage of Suggero was analyzed. This section presents the statistical results of the accuracy and details on the usage. With Suggero, it was possible to manually select some elements, choose a suggestion in Suggero to expand the selection, manually select additional strokes, and use Suggero again, and so on. To better understand people's strategies for using Suggero, a more detailed analysis of all Suggero trials was conducted. Trials in which no Suggero suggestions were used were classified into a no suggestions category, with remaining trials classified as high accuracy (1-3 interactions), medium accuracy (4-5 interactions) and low accuracy (6+ interactions). Among the high accuracy trials, trials with 1 manual selection + 1 suggestion were further classified as perfect accuracy. Participants were asked to use Suggero in 1080 trials (360 trials per *Complexity*). A k-independent samples t-test (Kruskal-Wallis) showed that these categories differ significantly ($p < .001$, $df = 2$) from each other. Interactions are either manually selected strokes or interaction with suggestion. Three interactions for example can mean two manually selected strokes and one selected suggestion or one manually selected stroke and two selected

suggestion. The interactions contain no informations about the order of inter-
actions. Figure 6.11 shows the breakdown of these categories. The reduction
of movement time and distance and the low number of interactions for *Sim-
ple* and *Challenging* can be credited to the high accuracy of Suggero in these
conditions. Being able to provide users with correct results is probably the
main goal of Suggero and any perceptual grouping system. The advantage is
not present in the *Arbitrary*, which was expected since the elements in the
target selections were not perceptually related.
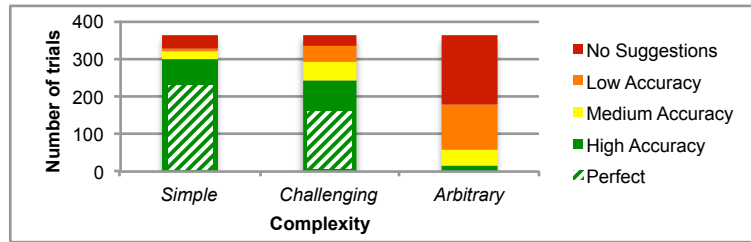
Figure 6.11: Accuracy of Suggero by *Complexity*.

6.8.6   Observations and Participant Ratings

A series of Wilcoxon Signed-Rank tests were used to compare ratings be-
tween techniques on the post-condition questionnaire. For *Arbitrary*, Har-
poon ($Mdn = 6$) was ranked significantly better than Suggero ($Mdn =
3, z = 3.532, p < .001$). There was no significant difference in ratings for
Simple (Suggero $Mdn = 5$; Harpoon $Mdn = 6$) and *Challenging* (Suggero:
$Mdn = 6$; Harpoon: $Mdn = 5$), as seen in Figure 6.12.
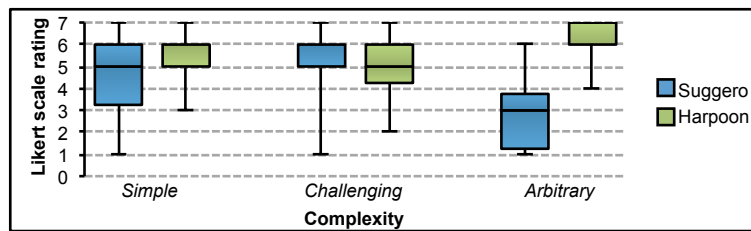
Figure 6.12: Participants rating on the question if they would use Suggero
or Harpoon per *Complexity* (1: never, 7: always).

Participant feedback was generally consistent with the ratings. Partic-
ipants understood and could use Suggero well for *Simple* and *Challenging*
complexity. For *Arbitrary*, participants reported difficulties performing se-
lections, with some appearing frustrated because they could not create the
correct selections with Suggero. Suggero was found to be helpful when cre-

ating complex selections. Two participants reported having difficulties identifying the content of the suggestions in the sketch because they were scaled and only showed suggested elements. Both people mentioned that indicating surrounding elements would be helpful. One participant mentioned that displaying more than three suggestions would be helpful (6 to 8), since he was able to identify at least that many possible groupings for a certain element. In the second phase of the study, participants could express their feedback verbally to the experimenter while drawing and performing selections. Participants tended to select semantically related elements (e.g. the car, house or person they added to the provided sketch). Also, participants tried to determine if their understanding of Suggero was correct by selecting elements sharing the same properties like color or shape. No participant tried to create arbitrary selections of perceptually unrelated objects. When participants were unable to perform their intended selection with Suggero, they often reported comments such as "that was too complex for it" or "I will try it [Suggero] for something simpler". This further indicates that participants were aware of Suggero functionalities, advantages, and limitations.

6.8.7   Hypotheses Discussion

Analysis showed that Suggero showed the intended results for *Simple* and *Challenging* in a majority of the cases. For *Arbitrary*, Suggero was not able to return desired groupings, which was expected and intended. This means, *Hypothesis 1* (correct suggestions with 1 or 2 manual selections) can be partly confirmed.

    Hypothesis 2 stated that Suggero will perform better for complex groups and Harpoon for simpler. For the *Simple* condition, Suggero outperformed Harpoon in terms of movement time and movement distance. Expectedly, Harpoon outperformed Suggero in terms of trial completion time. The interaction count for *Simple* was lower for Harpoon.

    Hypothesis 3 dealt with visually complex sketches and stated that Suggero will outperform Harpoon for target selection of perceptual groups (*Challenging*) and Harpoon will win for perceptually arbitrary selection (*Arbitrary*). Suggero outperformed Harpoon in terms of movement time and distance as well as interaction count for the *Challenging* condition. Harpoon outperformed Suggero for all these measurements for the *Arbitrary* condition. This means, *Hypothesis 3* can be confirmed.

6.8.8   Results Summary

Even though Suggero required higher trial completion time, breaking the results down revealed that it required less movement time and distance. This has promising implications for avoiding effects of fatigue, which is particularly important on large wall displays. A detailed analysis showed that these

benefits arise from requiring fewer interactions to perform the target selections. One important takeaway from these results is that Suggero has the potential to support certain types of selections. These selections are more difficult for manual selection tools like Harpoon because of their selection complexity while still being perceptual groups. Participants were aware of this, and feedback and observations showed that they were able to predict the behavior and success rate of Suggero. This finding is consistent with people's behavior in other applications like Adobe Photoshop or GIMP. Supportive selection tools like the Magic Wand have a very specific use case, and people understand that. People tend to use manual selection tools for very simple and very complex groups, but for perceptually related elements, Suggero is a good addition to the user's toolset. Omitting incorrect suggestions is as important as providing suggestions, as observations from the *Arbitrary* condition showed. Providing people with suggestions instead of trying to automatically select groups is an important step to avoid distraction or confusion. Without being able the read minds, providing suggestions can be a good way to help people make selections more easily and effortlessly.

Chapter 7

Conclusion and Future Work

In this work, a novel perceptual grouping tool called Suggero was presented. Suggero can assist with perceptually related selections in hand-drawn digital sketches by analyzing the content and suggesting possible completions. This work described the concepts and implementation of Sugerro in detail. A background of influential perceptual psychology was provided. The algorithms used to extract perceptual features and the dynamic grouping algorithm were presented. A preliminary study was conducted to gain insights into expected behavior with Suggero, and a second study showed that Suggero was able to make selections more effortless and decrease interactions and stylus movement. These factors are important to decrease effects of fatigue—a well-known problem on large, digital displays. Both studies were presented in detail, including results and their discussion.

Contribution

Since every interaction (like movement, rotation or recoloring) with elements of a sketch requires prior selection, the need for easy and effortless selection methods is high. Humans easily perceive visually connected elements in a sketch, but cannot take advantage of this ability when performing selections. This work addresses this issue. Suggero uses a combination of perceptual features and a novel usage of the hierarchical agglomerative clustering technique to find perceptual groups.

The contributions of this work include providing detailed informations about the algorithms used in Suggero. To evaluate Suggero, two experiments were conducted, including a novel approach to test perceptual grouping systems by using abstract drawings to control several levels of complexity. Insights into the possibilities and limitations of Suggero are provided as part of the contribution.

Future Work

Suggero is built to be context agnostic. This means that no information about sketches and its elements concerning context is provided or used. The perceptual groupings in a sketch may change if users know its context. Gestalt theory referred to this behavior in the law of past experience [10, 44]. Adding domain knowledge would be interesting in terms of perceptual grouping. Since the weights used in the algorithms are empirically determined, applying a learning algorithm to learn the weights for different domains would be one possibility to achieve this domain knowledge. In terms of interaction, a more detailed research on the presentation of the suggestion provided by Suggero should be conducted. Since the usage of the suggestions vary over different levels of visual and selection complexity (as shown in the second experiment), only providing users with high accuracy or perfect suggestions would improve the usability of the system. For the Feature Extraction, adding more element features like multi-element similarity and user selection gesture to infer the perceptual groups can be investigated in future work.

Appendix A

Tasks Experiment 2

A.1   Complexity Simple

This section contains the base drawing for the *Complexity* condition *Simple* (see Figure B.2) and all selection tasks (see Figure A.2 and A.3).
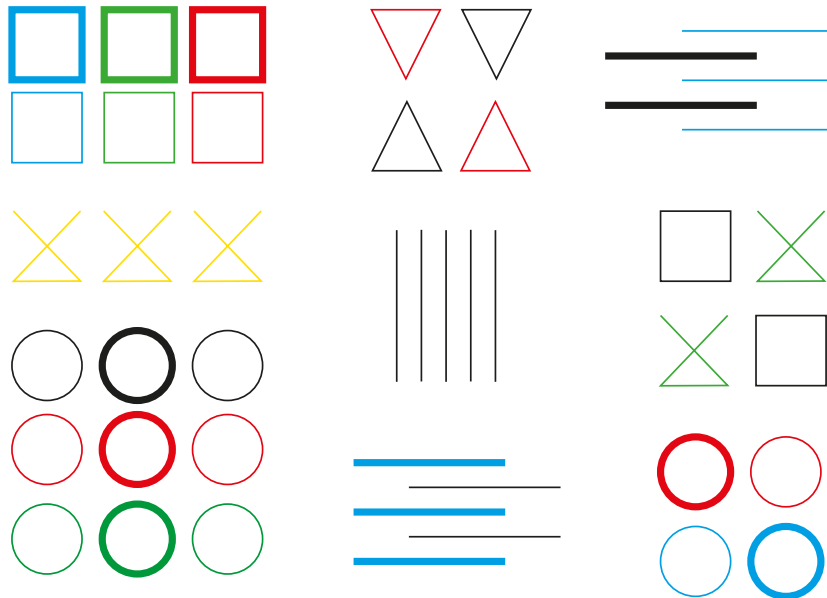
Figure A.1: Base drawing for *Complexity* condition *Simple*.
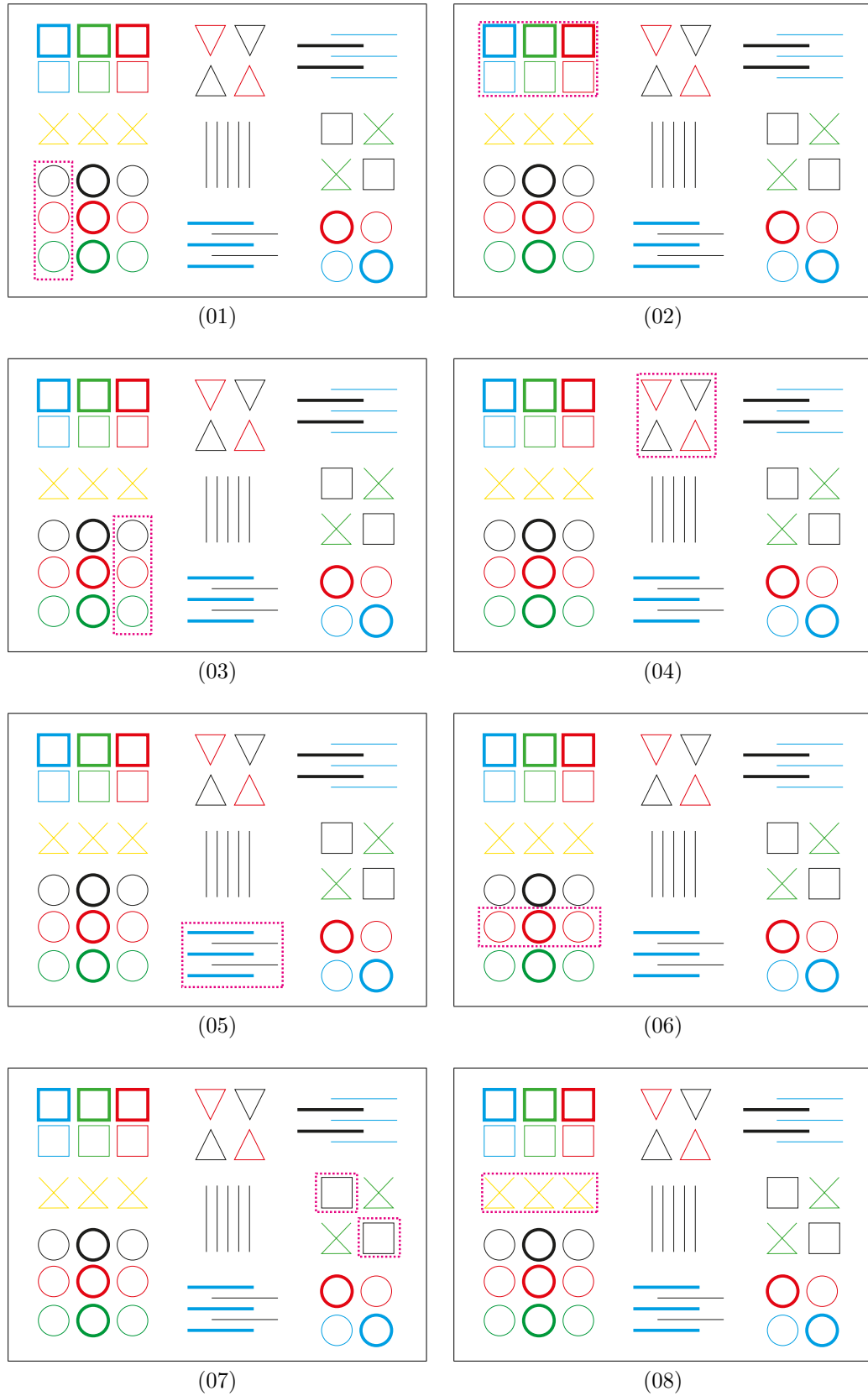
Figure A.2: Selections for *Simple* 1 - 8.

(09)

(10)

(11)

(12)

(13)

(14)

(15)

(16)
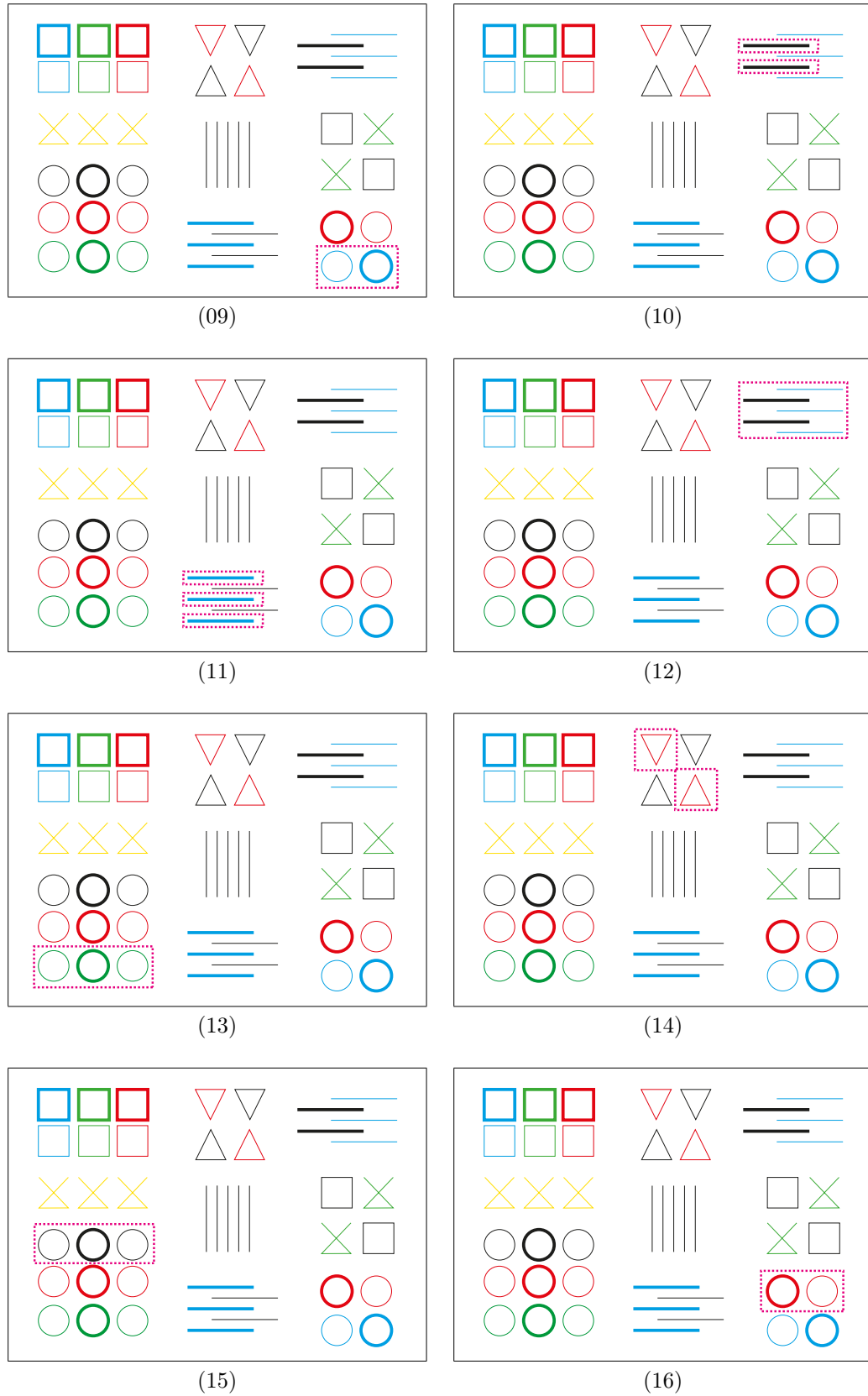
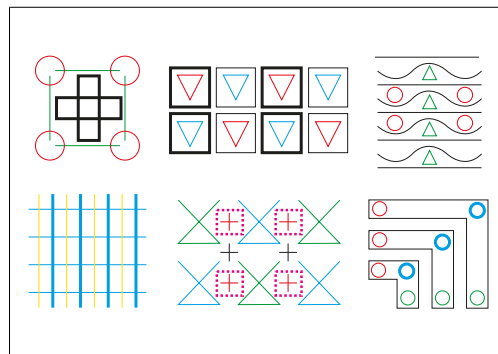**Figure A.3:** Selections for *Simple* 9-16.
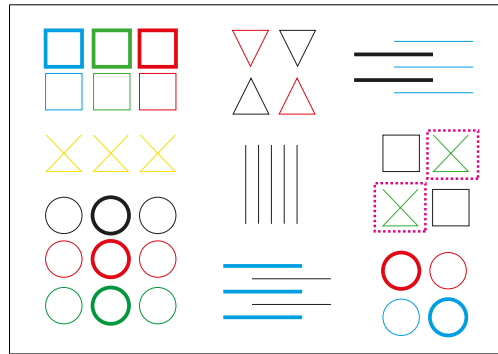
(17)
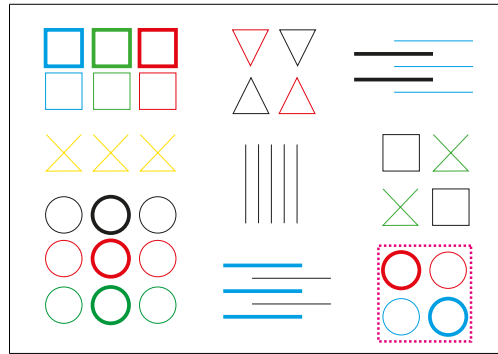
(18)
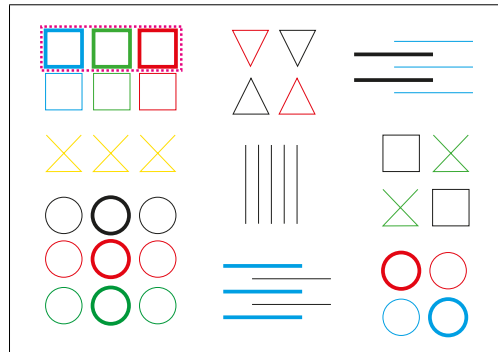
(19)
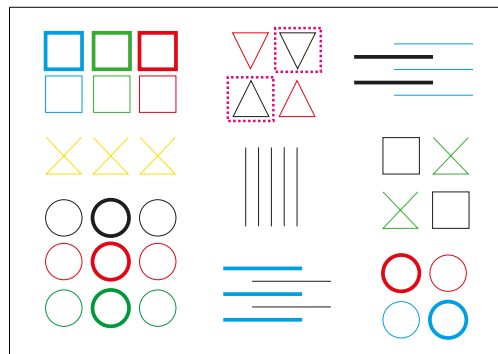
(20)

Figure A.4: Selections for *Simple* 17-20.

A.2   Complexity Challenging

This section contains the base drawing for the *Complexity* condition *Challenging* (see Figure A.5) and all selection tasks (see Figure A.6 and A.7).
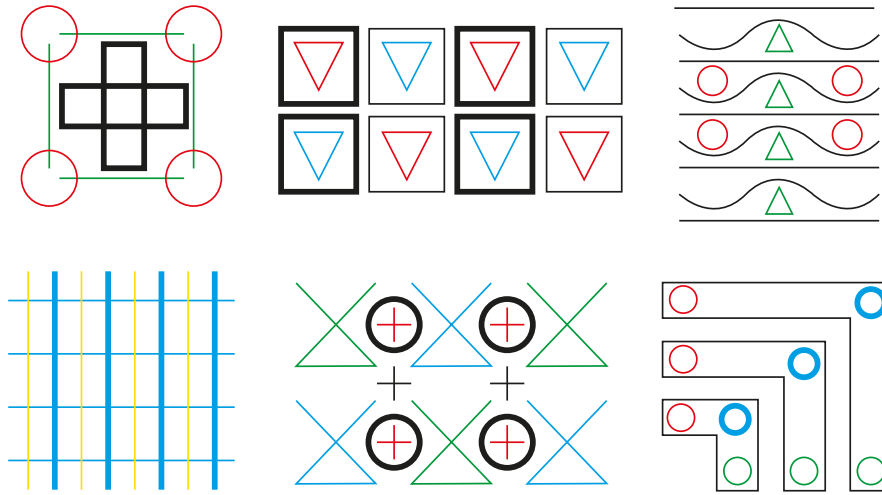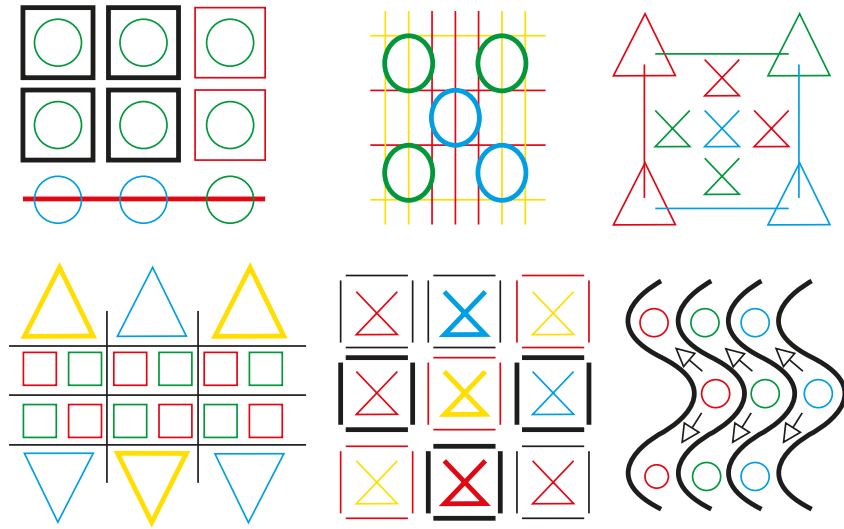

Figure A.5: Base drawing for *Complexity* condition *Challenging*.

(01)

(02)

(03)

(04)

(05)

(06)

(07)

(08)
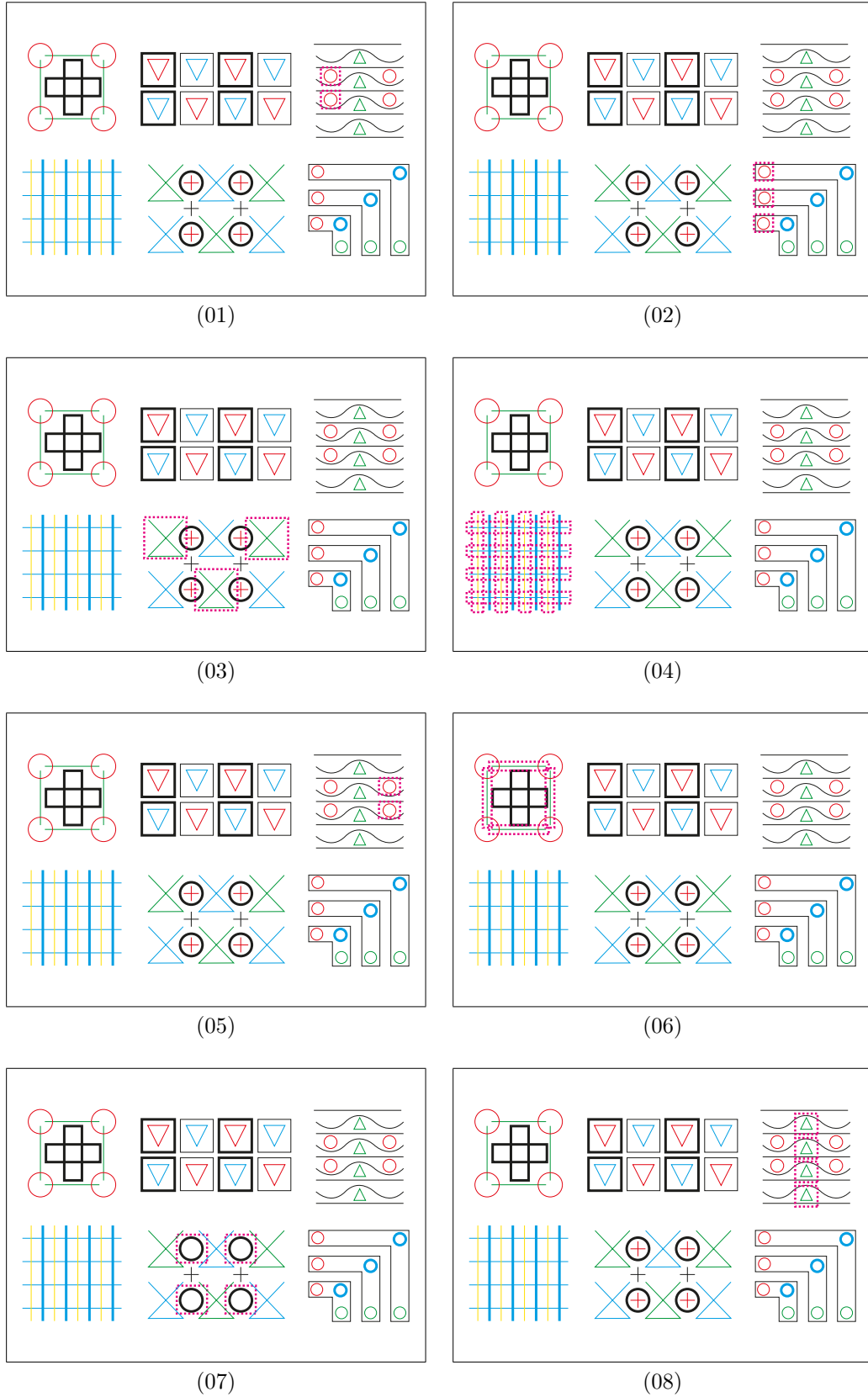
Figure A.6: Selections for *Challenging* 1 - 8.

(09)

(10)

(11)

(12)
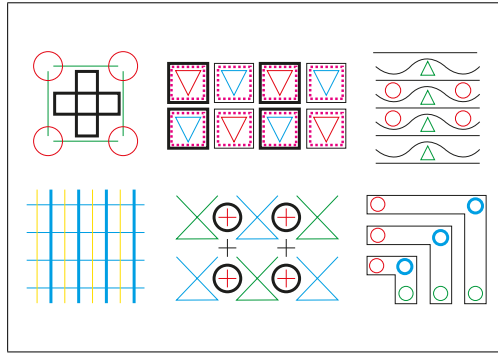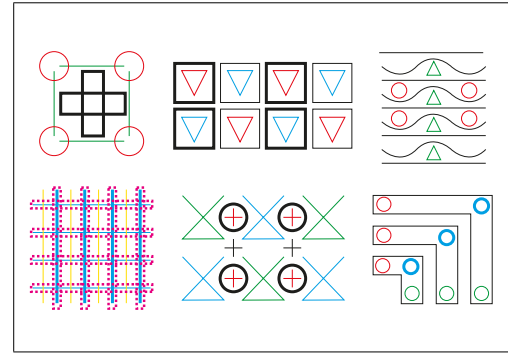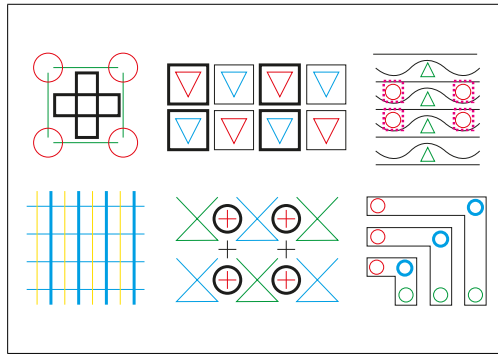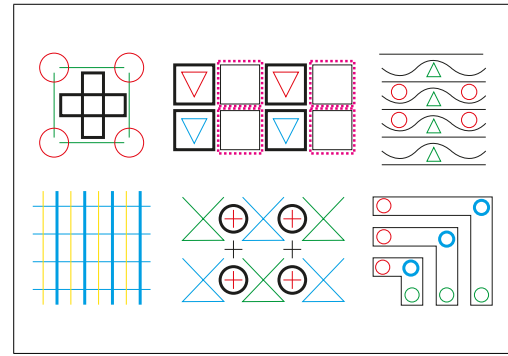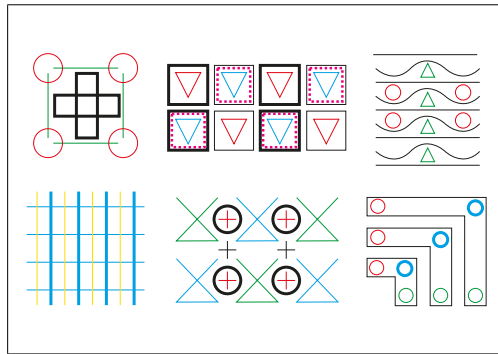
(13)
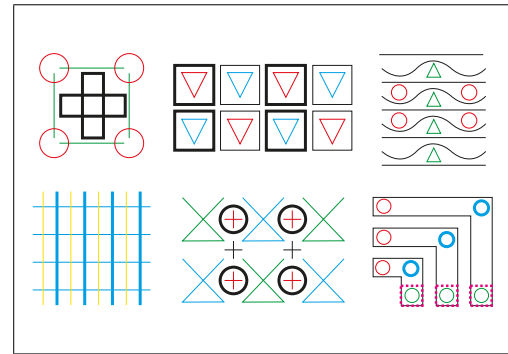
(14)
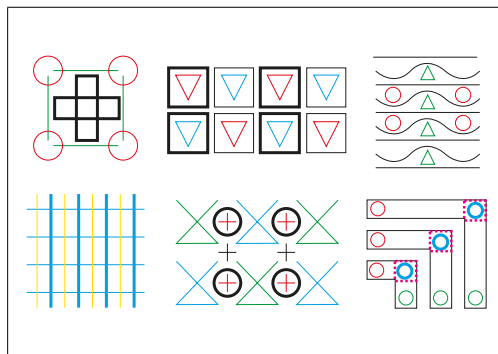
(15)
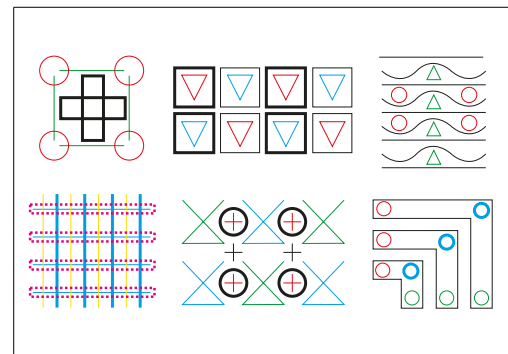
(16)

Figure A.7: Selections for *Challenging* 9-16.
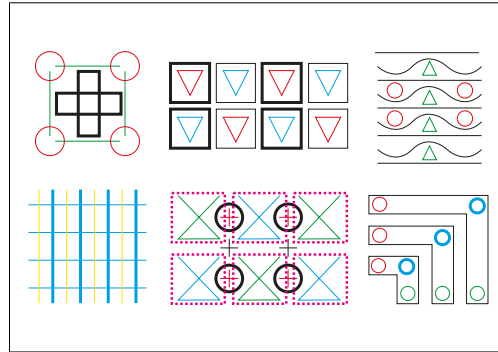
(17)
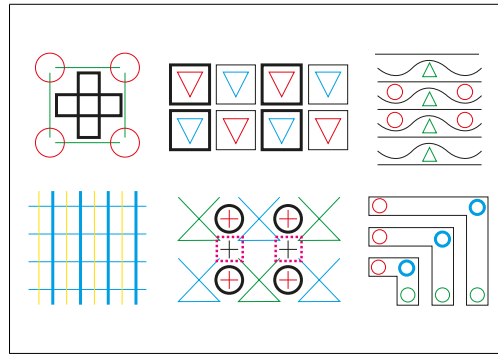
(18)
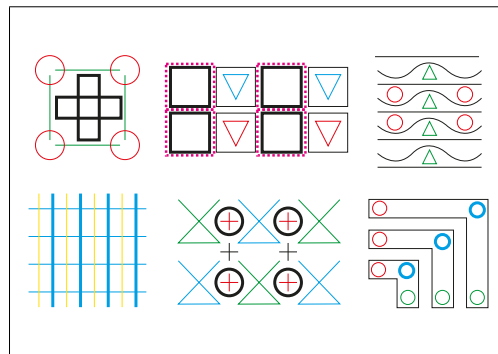
(19)

(20)

Figure A.8: Selections for *Challenging* 17-20.

A.3   Complexity Arbitrary

This section contains the base drawing for the *Complexity* condition *Arbitrary* (see Figure A.9) and all selection tasks (see Figure A.10, A.11, A.12).

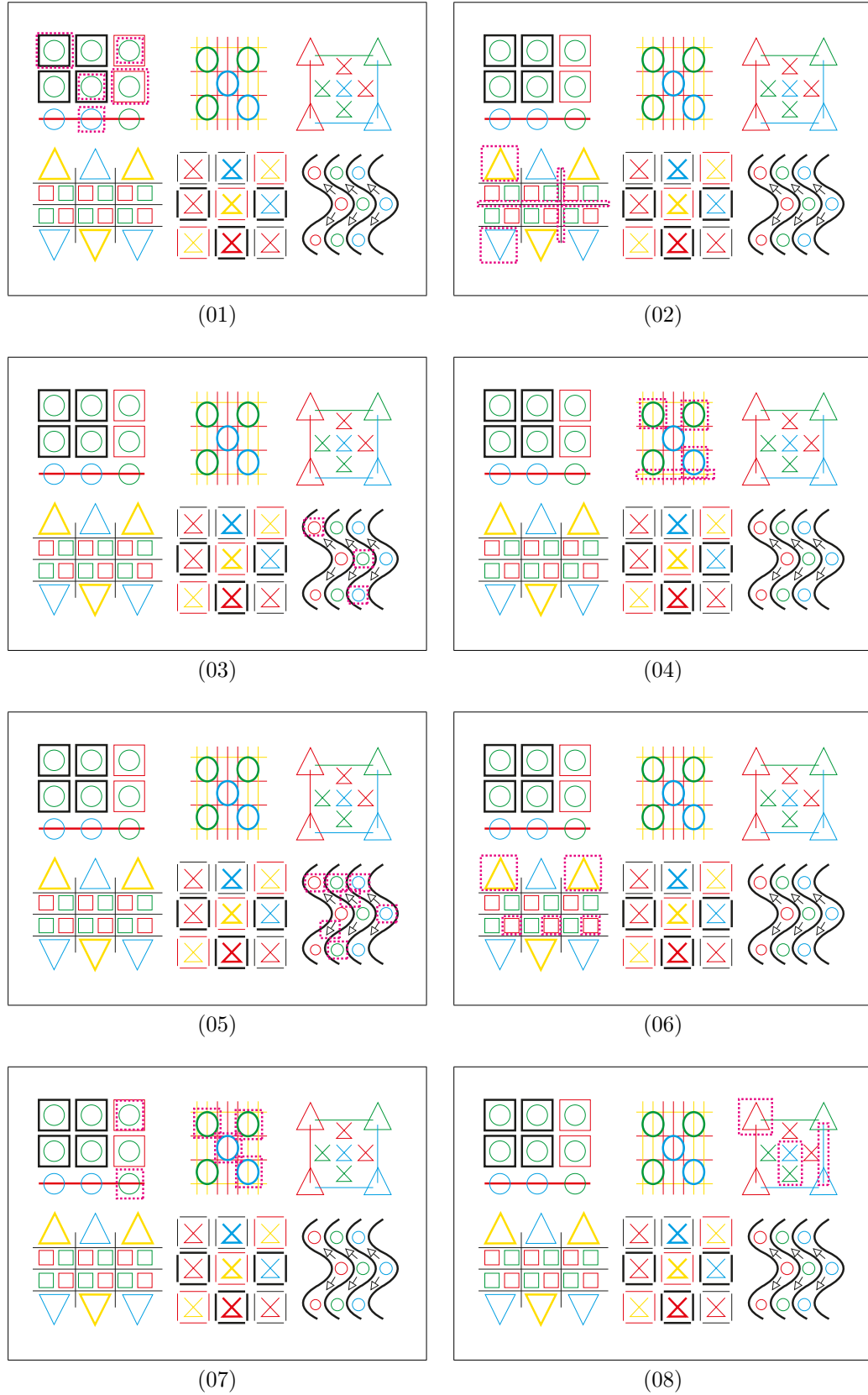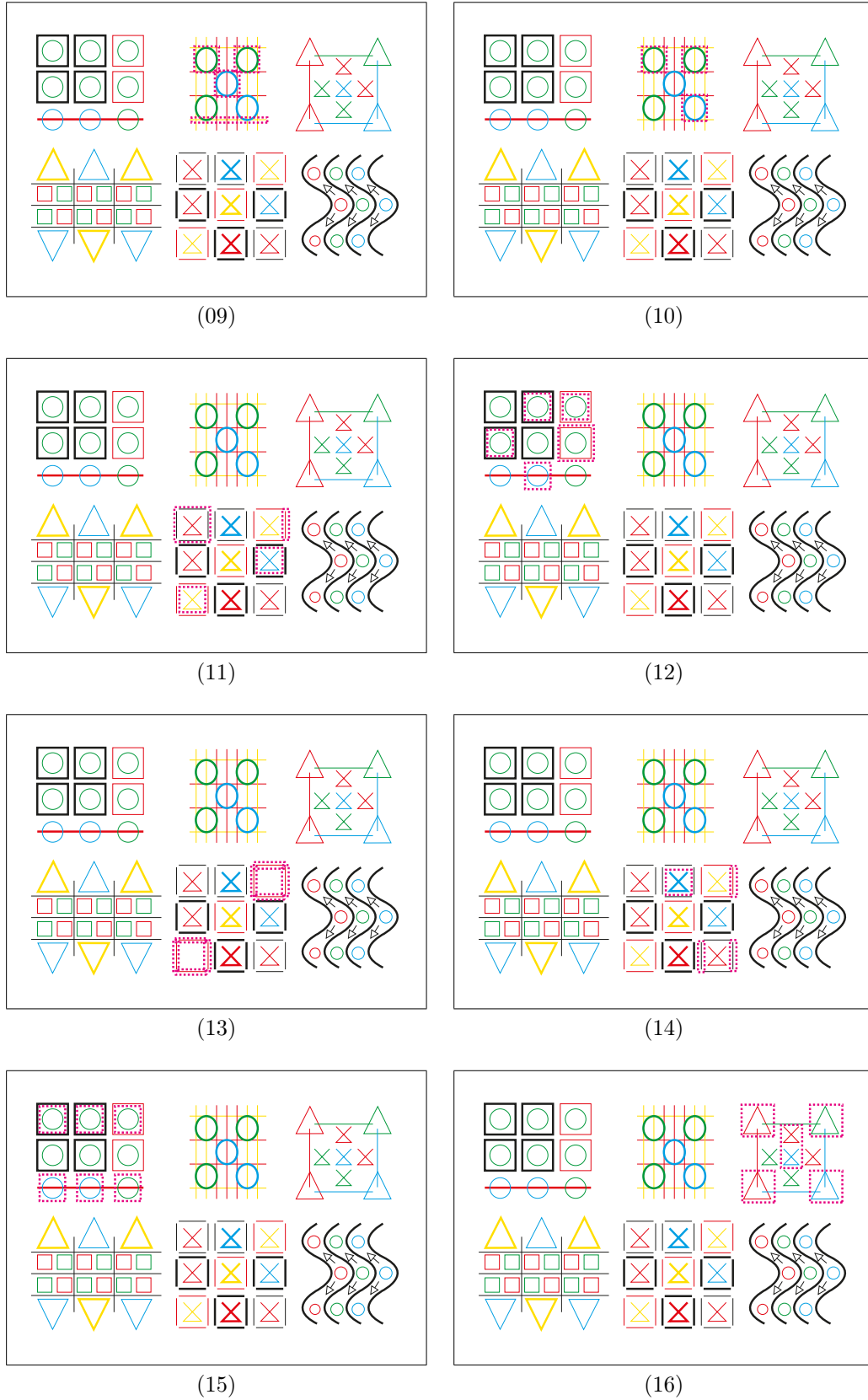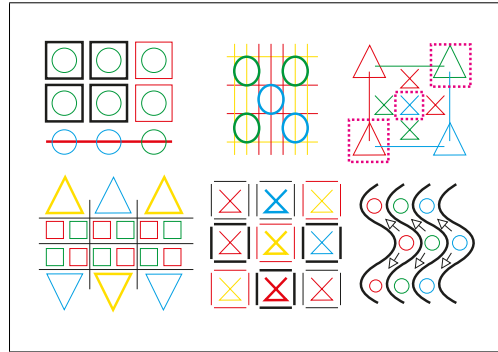**Figure A.9:** Base drawing for *Complexity* condition *Arbitrary*.

(01)

(02)

(03)

(04)

(05)

(06)

(07)

(08)

Figure A.10: Selections for *Arbitrary* 1 - 8.

(09)

(10)

(11)

(12)

(13)

(14)

(15)

(16)

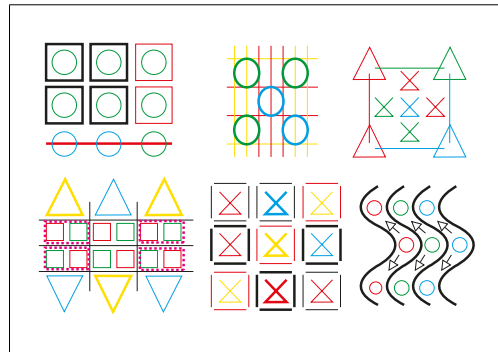**Figure A.11:** Selections for *Arbitrary* 9-16.
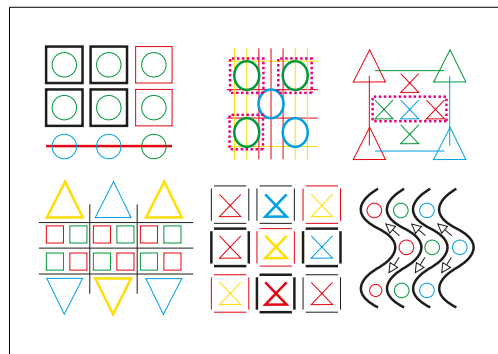
(17)
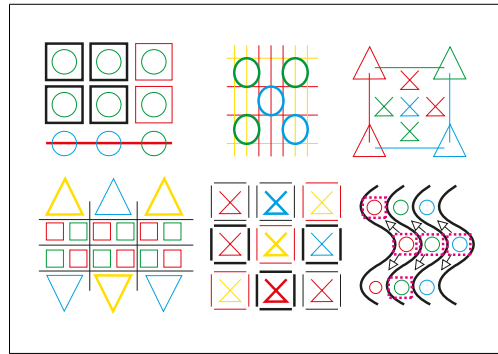
(18)

(19)

(20)

Figure A.12: Selections for *Arbitrary* 17-20.

Appendix B

Permissions

Permission to use co-authored work

September 25, 2012

I, Mark Hancock, give David Lindlbauer permission to use co-authored work from our paper

Lindlbauer, D., Haller, M., Hancock, M. and Scott, S. (2013). Perceptual Grouping: Selection Assistance for Digital Sketching. Submitted to *CHI '13: Proceedings of the 2013 annual conference on Human Factors in Computing Systems.*

for his master's thesis.

Sincerely,

Mark Hancock

Figure B.1: Permission to use co-authored work from Dr. Mark Hancock, University of Waterloo.
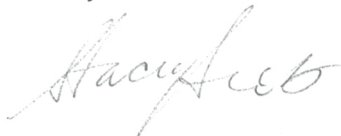
Permission to use co-authored work

September 25, 2012

I, Stacey Scott, give David Lindlbauer permission to use co-authored work from our paper

Lindlbauer, D., Haller, M., Hancock, M. and Scott, S. (2013). Perceptual Grouping: Selection Assistance for Digital Sketching. Submitted to *CHI '13: Proceedings of the 2013 annual conference on Human Factors in Computing Systems.*

for his master's thesis.

Sincerely,

Stacey Scott

Figure B.2: Permission to use co-authored work from Dr. Stacey Scott, University of Waterloo.

Appendix C

Contents of DVD

Format: DVD, Single Layer, ISO9660-Format

C.1   PDF-Files

Pfad: /

    Lindlbauer_David_2012.pdf Master's thesis

C.2   Miscellaneous

Pfad: /

    Experiment1_Tasks.zip   Tasks of first experiment
    Experiment2_Tasks.zip   Tasks of second experiment

Bibliography

Literature

[1]   Johnny Accot and Shumin Zhai. "Beyond Fitts' law: models for trajectory-based HCI tasks". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. Atlanta, Georgia, United States: ACM, 1997, pp. 295–302.

[2]   Fred Attneave. "Physical determinants of the judged complexity of shapes". In: *Quarterly Journal of Experimental Psychology* 53.4 (1957), pp. 221–227.

[3]   Fred Attneave and Malcolm D. Arnoult. "The quantitative study of shape and pattern perception." In: *Psychological Bulletin* 53.6 (1956), pp. 452–471.

[4]   Richard E. Bellman. *Dynamic Programming*. Princeton, USA: Princeton University Press, 1957.

[5]   Wilhelm Burger and Mark Burge. *Digitale Bildverarbeitung: Eine Einführung mit Java und ImageJ*. Berlin, Heidelberg: Springer, 2006.

[6]   Wilhelm Burger and Mark Burge. *Principles of Digital Image Processing – Advanced Techniques*. New York: Springer, 2012.

[7]   Jack Callahan, Don Hopkins, Weiser Mark, and Ben Shneiderman. "An empirical comparison of pie vs. linear menus". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press, 1988, pp. 95–100.

[8]   Sonya J. Cates. "Combining representations for improved sketch recognition". Ph.D Thesis. Cambridge, MA, USA: Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science., Sept. 2009.

[9]   Hoda Dehmeshki and Wolfgang Stuerzlinger. "Design and evaluation of a perceptual-based object group selection technique". In: *Proceedings of the 24th BCS Interaction Specialist Group Conference*. Dundee, United Kingdom: British Computer Society, 2010, pp. 365–373.
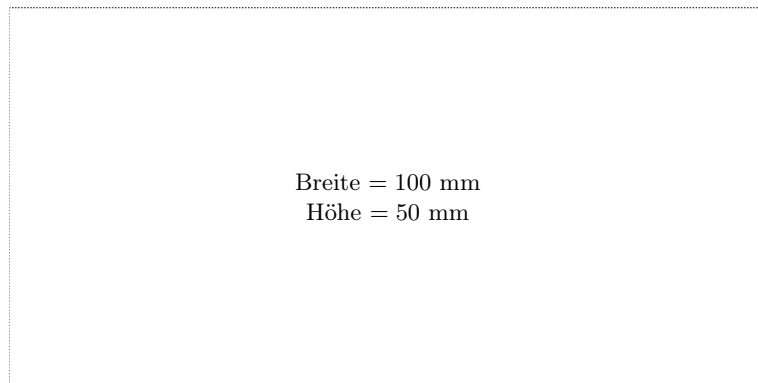
[10] Willis Davis Ellis. *A source book of Gestalt psychology*. London: Gestalt Journal Press,U.S., 1989.

[11] Leah Findlater and Joanna McGrenere. "A comparison of static, adaptive, and adaptable menus". In: *Proceedings of the 2004 conference on Human factors in computing systems*. Vienna, Austria: ACM, 2004, pp. 89–96.

[12] Paul M. Fitts. "The information capacity of the human motor system in controlling the amplitude of movement". In: *Journal of Experimental Psychology (General)* 47.6 (1954), pp. 381–391.

[13] Tovi Grossman, Patrick Baudisch, and Ken Hinckley. "Handle Flags: efficient and flexible selections for inking applications". In: *Proceedings of Graphics Interface 2009*. Kelowna, British Columbia, Canada: Canadian Information Processing Society, 2009, pp. 167–174.

[14] Tracy Hammond and Randall Davis. "LADDER: a language to describe drawing, display, and editing in sketch recognition". In: *ACM SIGGRAPH 2006 Courses*. Boston, Massachusetts: ACM, 2006.

[15] William Edmund Hick. "On the rate of gain of information". In: *Quarterly Journal of Experimental Psychology* 4 (1952), pp. 11–26.

[16] Ray Hyman. "Stimulus information as a determinant of reaction time". In: *Journal of Experimental Psychology* 45.3 (1953), pp. 188–196.

[17] Takeo Igarashi and John F. Hughes. "A suggestive interface for 3D drawing". In: *Proceedings of the 14th annual ACM symposium on User interface software and technology* Figure 1 (2001), pp. 173–181.

[18] Takeo Igarashi, Satoshi Matsuoka, and Toshiyuki Masui. "Adaptive recognition of implicit structures in human-organized layouts". In: *Proceedings of the 11th International IEEE Symposium on Visual Languages*. Washington, DC, USA: IEEE Computer Society, 1995, pp. 258–266.

[19] Daniel Kahneman and Avishai Henik. "Effects of Visual Grouping on Immediate Recall and Selective Attention". In: *Attention and Performance VI*. (Stockholm, Sweden). Hillsdale, New jersey: Lawrence Erlbaum Associates, July 1977, pp. 307–332.

[20] John F. Kelley. "An iterative design methodology for user-friendly natural language office information applications". In: *ACM Transactions on Information Systems* 2.1 (Jan. 1984), pp. 26–41.

[21] Benjamin King. "Step-Wise Clustering Procedures". English. In: *Journal of the American Statistical Association* 62.317 (1967), pp. 86–101.

[22]   Gordon Kurtenbach and William Buxton. "The limits of expert performance using hierarchic marking menus". In: *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. New York, NY, USA: ACM Press, 1993, pp. 482–487.

[23]   Gordon Kurtenbach and William Buxton. "User Learning and Performance with Marking Menus". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press, 1994, pp. 258–264.

[24]   Jakob Leitner and Michael Haller. "Harpoon selection: efficient selections for ungrouped content on large pen-based surfaces". In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. Santa Barbara, California, USA: ACM, 2011, pp. 593–602.

[25]   David G. Lowe and Thomas O. Binford. "Perceptual Organization as a Basis for Visual Recognition". In: *Proceedings of AAAI*. Washington, D.C, 1983, pp. 255–260.

[26]   Scott I. MacKenzie. "Fitts' Law as a Research and Design Tool in Human-Computer Interaction". In: *Human-Computer Interaction* 7.1 (1992), pp. 91–139.

[27]   Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2008.

[28]   Sachi Mizobuchi and Michiaki Yasumura. "Tapping vs. circling selections on pen-based devices: evidence for different performance-shaping factors". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. Vienna, Austria: ACM Press, 2004, pp. 607–614.

[29]   Matt Notowidigdo and Robert C Miller. "Off-line sketch interpretation". In: *Proceedings of AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural* (2004), pp. 120–126.

[30]   Aude Oliva, Michael L. Mack, Mochan Shrestha, and Angela Peeper. "Identifying the Perceptual Dimensions of Visual Complexity of Scenes". In: *Annual Conference of the Cognitve Science Society*. 2004, pp. 1041–1046.

[31]   Stephen E. Palmer. "Common region: A new principle of perceptual grouping". In: *Cognitive Psychology* 24.3 (1992), pp. 436–447.

[32]   Stephen E. Palmer. *Vision Science: Photons to Phenomenology*. 1.Edition. Cambridge, Massachusetts: Bradford Books, 1999, p. 832.

[33]   Stephen E. Palmer and Irvin Rock. "Rethinking perceptual organization: The role of uniform connectedness." In: *Psychonomic Bulletin & Review* 1.1 (1994), pp. 29–55.

[34] Brandon Paulson and Tracy Hammond. "PaleoSketch: accurate primitive sketch recognition and beautification". In: *Proceedings of the 13th international conference on Intelligent User Interfaces.* Gran Canaria, Spain: ACM, 2008, pp. 1–10.

[35] Dan Pelleg and Andrew W. Moore. "X-means: Extending K-means with Efficient Estimation of the Number of Clusters". In: *Proceedings of the Seventeenth International Conference on Machine Learning.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 727–734.

[36] Erich Rome. "Simulating Perceptual Clustering by Gestalt Principles". In: *Proceedings of 25th Workshop of the Austrian Association for Pattern Recognition.* Citeseer, 2001, pp. 191–198.

[37] Eric Saund, James V Mahoney, David Fleet, Daniel Larner, and Edward Lank. "Perceptual organization as a foundation for intelligent sketch editing". In: *AAAI Spring Symposium on Sketch Understanding.* 2002, pp. 118–125.

[38] Frank M. III Shipman, Catherine C. Marshall, and Thomas P. Moran. "Finding and Using Implicit Structure in Human-Organized Spatial Layouts of Information". In: ACM Press, 1995, pp. 346–353.

[39] Moshe Shpitalni and Hod Lipson. "Classification of Sketch Strokes and Corner Detection Using Conic Sections and Adaptive Clustering". In: *ASME Journal of Mechanical Design* 119 (1996), pp. 131–135.

[40] Kristinn R Thórisson. "Simulated Perceptual Grouping : An Application to Human-Computer Interaction". In: *Cognitive Science* (1994), pp. 876–881.

[41] Anne Treisman. "Features and objects in visual processing". In: *Scientific American* 255.5 (1986), pp. 114–125.

[42] Anne Treisman. "Perceptual grouping and attention in visual search for features and for objects." In: *Journal of Experimental Psychology: Human Perception and Performance* 8.2 (1982), pp. 194–214.

[43] Anne Treisman and Garry Gelade. "A feature-integration theory of attention". In: *Cognitive Psychology* 12.1 (1980), pp. 97–136.

[44] Max Wertheimer. "Untersuchungen zur Lehre von der Gestalt. II". In: *Psychological Research* 4 (1 1923). 10.1007/BF00410640, pp. 301–350.

[45] Markus Wuersch and Max J. Egenhofer. "Perceptual Sketch Interpretation". In: *13th International Symposium on Spatial Data Handling.* Montpellier, France: Springer–Verlag, 2008, pp. 19–38.

[46] Dengsheng Zhang and Guojun Lu. "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures". In: *Intelligent Multimedia and Distance Education.* 2001.

Messbox zur Druckkontrolle

Breite = 100 mm
Höhe = 50 mm