in: Second International Conference on Theory and Application of Diagrams (Diagrams 2002), Springer, Berlin, 2002.

ViCo: A Metric for the Complexity of Information Visualizations

Johannes Gärtner[1], Silvia Miksch[2], and Stefan Carl-McGrath[3]

[1] Institute of Design and Assessment of Technology
Vienna University of Technology, Argentinierstraße 8/E 187, A-1040 Vienna, Austria
johannes.gaertner@tuwien.ac.at
http://time.iguw.tuwien.ac.at/JOG

[2] Institute of Software Technology and Interactive Systems
Vienna University of Technology, Favoritenstraße 9-11/E 188, A-1040 Vienna, Austria
silvia@ifs.tuwien.ac.at
http://www.ifs.tuwien.ac.at/~silvia

[3] Otto-von-Guericke University, Department of Simulation and Graphics
P.O. Box 4120, D-39016 Magdeburg, Germany
stefan.carl@student.uni-magdeburg.de
http://www.uni-magdeburg.de/carl/

Abstract. Information Visualization produces a visual representation of abstract data in order to facilitate a deeper level of understanding of the data under investigation. This paper introduces *ViCo*, a metric for assessing Information Visualization complexity. The proposed metric allows for the measurement of Information Visualization complexity with respect to tasks and users. The algorithm for developing such a metric for any chosen collection of visualizations is described in general and then applied to two examples for purposes of illustration.

1 Introduction

Within the field of visualization, Information Visualization aims for supporting individuals in understanding and detecting the relevant features of a field of interest. Information Visualization is the use of computer-supported, interactive, and visual representations of abstract data to facilitate cognition. The goal of Information Visualization is to ease understanding, promote a deeper level of understanding of the data under investigation, and foster new insight into underlying processes. The fields of application may vary from scientific tasks to everyday purposes.

Important contributions to the field of Information Visualization have come from various directions. In his seminal books on visualization Edward Tufte ([16], [17], [18]) discussed a number of interesting visualizations. He also introduced a number of recommendations for the design of such graphics (e.g., removing elements that do not contain information, minimizing gray) and a number of interesting concepts to enhance visualization design and analysis (e.g., macro and micro reading, which enables readings of a visualization at various levels of abstraction and detail).

Information Visualization covers a broad field of visualizations and a number of books try to present relevant knowledge on dos and don'ts of specific design elements (e.g., [3], [14], [15]).

In many cases there is broad consensus on whether a specific visualization is good or not good. However, there is little theory to support such judgment. One way to deal with this situation is to develop benchmarks for the evaluation of visualizations where standardized sets of data and tasks are visualized in various ways [8]. it is difficult to quantitatively measure visualizations and to understand when to apply one visualization compared to another. To address some of these limitations, we are focusing on the development of an appropriate metric, called *ViCo,* by which to judge visualization and Information Visualization in particular. Our approach takes into account the tasks to be accomplished and the users' knowledge and needs on the one hand, and on the other hand, the difficult procedure of quantifying qualitative information concerning what we call here cognitive elements or operations. Hence, the metric *ViCo* can be seen as an algorithm that allows a quantitative comparison of the relative complexity of a set of visualizations for any given situation.

For example, assume we have two visualizations and let #N be the number of items being represented. Then, keeping all other parts being equal, the representation that makes it necessary to read #N items once is substantially better than one that necessitates reading #N * #N items. We want to facilitate the development of such formulas and comparisons.

In the next section, we develop the conceptual fundamentals of the *ViCo* algorithm. The algorithm itself is then introduced in the third section. The fourth section illustrates the algorithm through two examples. Finally, we discuss related issues and present concluding remarks.

The authors have experiences in task-specific approaches of Information Visualization, which range from visualization for software development and management consulting [5], to visual representations for various monitoring data and processes of patients in intensive care units ([10], [12]) and for the design of shift-rotas in various industries [6].

2 Conceptual Fundamentals

Here we present central conceptual fundamentals and definitions, which are needed to proceed with our approach and then are heavily exploited in the later algorithm we develop in section 3. These definitions include (1) reading and writing, (2) comparisons and calculations, (3) tasks and users, (4) complexity, and (5) the metric.

- **Reading and Writing**

Berg [2], inspired by actor-network theory and work within Computer Supported Cooperative Work (CSCW), tries to circumvent technological-determinist as well as social-constructivist accounts in discussing the changes brought about by the use of artifacts. He aims for a relational conceptualization of what such tools do, without attributing the activities exclusively to the tool itself or to the person working with it. He conceptualizes the activities associated with information technology in work practices as *reading* and *writing* of artifacts. This enables a consistency of approach in analyzing the paper-based and computer-based technologies. For his field of analysis – electronic patient records – he describes the generative power of artifacts as accumulating inscriptions and coordinating activities, thus making the handling of more complex work tasks possible.

Transferring this conceptualization of computer artifacts to the field of Information Visualization, a first element of complexity comparisons will relate to such reading and writing of visualization. Specifically, how many things do users have to read or to write for a given visualization?

- **Comparisons and Calculations**

Expanding on Berg's [2] approach (i.e. considering the use of artifacts in terms of the cognitive activities of users) one has to consider other activities that might be of relevance for Information Visualization. Two additional activities are considered here: *comparisons* and *calculations*. Comparisons deal with comparing one or more elements of Information Visualizations with respect to specific features. Calculations may influence the task or problem processing in two ways: first, that something can be computed (compare [2]) and, second, the effort of computing may vary [5].

- **Tasks and Users**

Two critical elements are missing so far: *tasks* and *users*.

It is impossible to discuss the amount of reading, and writing, comparing, and calculating that is necessary without specifying a task and supposing a user up front. Only when it is clear whether a task is completed or not can one discuss the amount of reading, writing, comparing, or calculating that is necessary.

Information Visualization complexity can only be discussed with respect to the same tasks. Similar constraints are described in the field of designing maps. For example, MacEachren [11] argues that there cannot be a discussion of how good or bad a map is without knowledge on the various ways of its use.

Users are to be considered too. The analysis of visualization complexity cannot be conducted without some reference to the users of a given visualization. Reason is that the information users can gain by using an Information Visualization depends also on their general and task-specific knowledge (e.g., to interpret graphics on various accounting measures, one must understand the categories of accounting; to understand the tableau of chemical elements, one must know something about chemistry).

- **Complexity**

We conceptualize the complexity of visualizations in terms of the operations – or cognitive elements - needed to accomplish the tasks by users. This approach relates strongly to the field of computational complexity [13], a part of computer sciences.

The proposed metric of complexity will not deliver a single number but will describe a function with various variables (e.g., number of items to be compared). For example, a simple algorithm for finding the median of n items uses $k*n \log n$ comparisons, (where k is a constant of proportionality). Here "n" is a "*variable*". In computing, it is common to use the "size of the input" as the main variable. In our case, we use variables to denote the different dimensions of input, which are relevant to comprehend the visualization. Additionally, complexity analysis in computer science provides both *upper* and *lower bounds*. For example, median finding has a lower bound of $2n$ comparisons (a proof that any algorithm for median finding must make at least this many comparisons), and an upper bound of about $3n$ (worst-case runtime of the best algorithm for median finding) [1]. For our approach, it would also be interesting to consider upper and lower bounds for visualization tasks.

In our case, the necessary variables may be difficult to identify and the number of variables considered is expected to spread over time, as the analysis of a specific field of Information Visualization matures and deepens. Though a function is more difficult to handle than a single number, a function seems an appropriate way for the comparison of visualization. For instance, researcher and designers can gauge which visualization to use under what circumstances. Furthermore, it is not unusual to work with functions to describe complexity. Again, computational complexity within computer sciences works strongly with such elements.

- **The Metric**

Science distinguishes a number of ways to compare or describe features of objects of interest. From a mathematical point of view, the highest level of such comparisons leads to scalar, absolute values. On a level lower, observers would agree on the ordering and relative distances of complexity (e.g., 1-2-4; 3-6-12), or even weaker ordering function (e.g., A > B, B > C).

As mentioned in the previous section we do think that the computation of complexity relies on defining tasks to be accomplished by users. It would be too much to expect the metric (and its procedures) to guarantee that its users of the metric reach consensus on which tasks and user groups to take as the starting point. However, we consider it plausible – and will discuss it later on – that it should be possible to come up with a list of relevant tasks in close to all situations and to articulate reasonable assumptions with respect to the users. Afterall, Information Visualization typically makes use of information that already refers to such tasks and user groups.

Under the condition of shared assumptions regarding users and tasks, the metric we develop will be able to compute the complexity of Information Visualizations on a particular level.

3 Our Approach: A Metric for the Complexity of Visualizations

In the following we will describe the proposed algorithm to develop the metric of complexity for a chosen set of visualizations, called *ViCo* (Visualization and Complexity). We first describe the steps of the algorithm and then show their application on two examples.

The algorithmic steps of *ViCo* are:

1. Analyze the tasks to be accomplished by the use of a set of given visualizations and select those tasks to be taken as the basis of measurement.
2. Define minimal reading, writing, comparing, and calculating operations with respect to users' groups and variables of the data set to be visualized.

3. Develop the functions that describe the number of such operations needed to accomplish such a task.

We make the assumption here that the visualizations under consideration include all the information necessary to complete the tasks at hand. Though similar visualizations ([6], [10]) may vary substantially in what tasks they allow one to work on, this line of inquiry shall not be pursued here, because we are focusing on approaches which stay as simple as possible to communicate complex data and information in diagrammatic form.

3.1 Tasks

The first step of *ViCo* is to define the tasks that are the basis of the later measurement. In many cases this selection will be straightforward. For example,

- Understanding differences between object A and object B,
- Finding an object, or
- Being able to decide whether something is true or false.

In other cases, with a large number of tasks, a selection process may be needed. In most cases it should be possible to come up with a reasonable number of the most relevant tasks or at least relevant examples of tasks. However, if developers of an Information Visualization have no idea about possibly relevant tasks that users will try to accomplish with such Information Visualization, we would recommend to do more exploration in that direction, before starting the work of visualization design and analysis.

After selecting tasks, a further refinement is needed. A task is defines as such for our further analysis if (and only if) we are able to determine whether it is completed or not, and this typically calls for further refinement:

- Understanding differences between object A and object B with respect to predefined quantity of features (e.g., all, some, a percentage, etc. of the features),
- Finding a particular object (e.g., the street within a map), or
- Being able to decide whether statements A, B, C are true or false.

Again we assume that such refinements should be possible in most or all practically relevant situations.

3.2 Reading & Writing, Comparing & Calculating, and Users & Variables

As mentioned in the second section, Berg [2] focused on reading and writing operations. As long as we deal with visualizations drawn by the computer there is little user-writing involved. However, if we take into account interactive parts of the Information Visualization process, then the tasks of writing and typing become a crucial part of the complexity analysis as well.

Besides reading and writing, we consider comparisons and calculations as separate operations. It seems possible to develop additional categories as well that might help to focus better on further activities (e.g., group processes). Our metric is open to such extensions.

In the following we explain how the reading, writing, comparing, and calculating operations are defined. At first glance this might look rather tricky. However, it is so only to some degree, as we go for *relative complexity* of visualizations and not for *absolute complexity*. We do not attempt to develop a metric that covers all possible visualizations, for all possible tasks for all possible user groups. We go for a smaller objective: We want to be able to compute the complexity for any given set of visualizations with given tasks and given assumptions regarding the user group. This allows for incremental enlargement for any specific field but avoids the pitfalls of a universalistic approach.

Looking closer at reading, writing, comparing, and calculating, the question arises at what level to measure these activities. It is possible to conceptualize these operations in extremely complex ways. Again, we go for a smaller aim. We try to find the simplest possible operations for a given set of visualizations.

When looking at simple conceptualizations, possible types of such operations could look like the following. The conceptualization of reading, comparing, and calculating, can be seen in analogy to the various levels of perception (see for example [7]). Writing we conceptualize as straightforward activity:

At least three levels of reading operations can be distinguished:

1. Operations with the eye (e.g., finding a legend)
2. Basic operations for reading a letter or a word; or finding the next row, etc.
3. Cognitive operations (e.g., memorizing)

At least two levels of comparison operations can be distinguished:

1. Direct comparison
2. Comparison with memorizing

At least two levels of calculation operations can be distinguished:

1. Actual calculation
2. Cognitive processes in order to develop a way how to calculate

It is not always necessary to work with the operations on the visual level. Dropping such measurement seems reasonable if no relevant differences can be expected between the visualizations to be analyzed. This might be the case if the operations defined (e.g., finding the start of row, finding a column) do not vary strongly between the visualizations at hand. If high differences between visualizations can be expected then measurement of eye movement should be done. Techniques to measure and compare such eye movement are used within usability labs (e.g., eye tracking), and the results of such measurements depict the time needed for a task or operation depending on relevant variables (e.g., number of columns). Statistical measures would then apply here.

The simplest measures – and those this article tries to exploit as far as possible – are simple operations of reading a letter or a word, comparing two lines, etc. Such operations should be selected on the highest possible level with respect to the visualizations under investigation. For example, if two visualizations both rely on bars and make it necessary to compare them, such basic operations could be: (a) Find pairs of bars that shall be compared, and (b) Compare two bars.

Using cognitive operations [7] as a foundation of measurements may sound unusual from the perspective of computer scientists. However, they are not as bad as one might expect. For reasons of measuring complexity, we can simplify dramatically by again defining basic cognitive elements (e.g., reading a word). These basic elements can be used without further clarification as long as they are used in the same way for all visualizations of interest in a situation.

For example, if the comparisons of numbers have to be made, such a "comparison" would be a cognitive minimal element. It would not make sense to go into further detail (e.g., understanding all the processes involved in such a comparison) as long as the minimal element meets the following requirements:

1. It is used consistently with respect to the visualizations at hand (*consistent*).
2. It does not vary internally in relevant ways (e.g., words in visualization D are dramatically shorter than words in visualization E) (*invariant*).
3. It does not overlap with other operations – either within or in between tasks or visualizations (i.e., if two operations are used that somehow overlap in their utility, they have to be split up in smaller operations) (*irreducible*).

If a cognitive element does not meet the above criteria then further refinement is needed. Such refinement typically brings in features of *users* (e.g., users do or do not know how to read a specific element of a representation) or additional *variables* (e.g., length of words). These variables may refer back to the task or to other features of the process, the visualization, etc.

Whenever decisions have to be made regarding the level of knowledge of users one can expect, this either brings in an additional variable or an assumption regarding the users that holds true for all visualizations under investigation. Knowledge of users may refer to general knowledge and capabilities or task-specific, situation-specific knowledge. It is important to understand that this does not call for a complete collection of all user knowledge. Only if a reading operation depends (in its feasibility or complexity) on specific knowledge will a decision have to be made about whether to assume that expected users will have that knowledge or to make a variable out of it (compare the explanations about variables used in computational complexity in section 2). The first approach simplifies the function but limits its applicability. The second approach increases the scope of applicability of a comparison to more user groups. However, this comes at the price of higher complexity of the function. We are aware that such a set of assumptions regarding users can be increased indefinitely. From a practical point of view however, the number of elements to be added will depend on the interactions of those persons involved in developing the metric. Therefore, the list should be limited, but open for later amendments.

Again, if designers of visualizations do not have an idea about their users, it seems worthwhile to think about this. In most cases however it should be clear. If different basic operations lead to different complexity results, this indicates weaknesses or differences in these definitions.

Summing up, after defining a set of basic operations, the variables to be considered, and (some) assumptions regarding the knowledge of the expected users, we can start with the calculation. The variables of the complexity function are a side product of the above analysis.

3.3 Develop the Functions to Compute the Number of Such Operations

After defining basic operations and variables one should be able to describe the complexity of reading, writing, comparing, and calculating of Information Visualizations in terms of software programs. Such programs finish when the corresponding task is fulfilled.

Correspondingly, it is necessary to develop an algorithm that accomplishes the task with the operations defined. Then – with standard techniques of computer sciences – one can compute the complexity as a function of the variables introduced in a reproducible way.[1]

The results of this approach should be rather stable. Algorithms should not vary too strongly between applicants. A change of basic operations should only lead to a change in the resulting function if it introduces a new operation or a new variable. Both of these options are consistent with the metric.

The complexity of the algorithm (building upon well-defined tasks and well-defined operations) then is also the complexity of the visualization. To facilitate visualization comparisons, it may make sense to further simplify the functions describing the complexity. The basic operations used in these algorithms have constant time (they do not depend on variables!). Correspondingly, one operation can be described as a multiple of the other `Op1=a*Op2` by using scalars `a, b,` etc.

Summing up, a complexity analysis builds upon the elements listed in Table 1 (all necessary definitions were given in the previous section):

$$\begin{aligned}
Vis &= \{ Visualization_A,\ Visualization_B, \ldots \} \\
Task &= \{ Tasks\ to\ be\ achieved\ with\ Vis \} \\
User &= \{ Assumptions\ about\ the\ users\ e.g.,\ knowledge \} \\
Var &= \{ Variables\ used\ in\ at\ least\ one\ operation \} \\
Op &= \left\{ operations(Var) \;\middle|\; \begin{array}{l} \forall v \in Vis\ \exists\ algorithm\ to\ accomplish\ \forall t \in Task\ building\ upon\ these\ operations \\ operations(Var)\ is\ \text{consistent} \wedge \text{invariant} \wedge \text{irreducible} \end{array} \right\}
\end{aligned}$$

Table 1. Definitions of the various elements of the metric *ViCo*.

Using the described algorithm it is possible to develop the metric *ViCo* for any chosen set of visualizations and correspondingly compute the relative complexity of a set of visualizations. This measure of complexity relies on reasonable definitions of tasks, reasonable assumptions regarding users, well-defined operations, and variables describing features of the problems at hand that are considered in the assessment. In the next section, we illustrate *ViCo* with two examples.

---

1 We are aware of the fundamental limitations in this field (e.g., the question whether an algorithm is the simplest possible algorithm for the task to be accomplished cannot be solved in general). However, we expect most actual algorithms to be simple, because the building blocks of the algorithms – the operations – are complex. The complexity is in the operations and not in the overall algorithm. For example, it is very difficult for us to code good algorithms for face recognition, but people do this with ease. “Simple for the human brain” does not mean “simple for us to code as an algorithm”.

4 Examples

In this section we explain how our complexity metric, *ViCo,* is applied to the following two examples, (1) Tasting Whisky and (2) Visualizations of some issues regarding the Challenger's Disaster.

4.1 Example 1: Tasting Whisky

Tasting whisky is a very complicated task, which is done principally with the nose, then by the tongue, etc. The taste of Whisky can be graded in 10 categories on the scale of 0-3 for each (3 being the highest). If you use a star plot [4] (also called a wheel) each category corresponds to a spoke of the wheel. When you finish the grading and join up the lines, a particular shape of wheel appears, which reflects the characteristics of the Whisky. Figure 1, shows two examples: on the left-hand side the star plot of "The Balevenie, 12 years old" and on the right-hand side "Glenfiddich" (taken from http://www.scotchwhisky.com/).

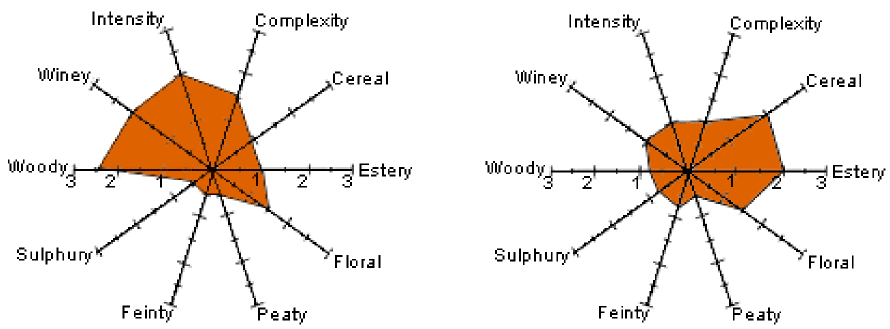


Fig. 1. Visualizing the taste of two types of Whisky. On the left-hand side, the star plot of "The Balevenie, 12 years old" and on the right-hand side, "Glenfiddich" (taken from http://www.scotchwhisky.com/).

In the following Table 2, the visualizations (Vis), the Tasks (T), the assumptions regarding the users (Users), the variables (Var) and then Operations (Op) are defined. After that the algorithms for accomplishing two tasks (compare Table 2) with the operations are described. Building upon that the complexity functions are developed.

Kind	**Name**	**Explanations**
`Vis1`	`Star Plot`	See above Figure 1
`T1`	`Highly Similar?`	Determine whether the whiskys under consideration have highly similar features
`T2`	`Identify Differences`	Find and identify main differences of the whiskys
`Users`		• are able to understand and read star plots • are familiar with the 10 features of whiskey

`Var`	`#C`	• number of categories/spokes
`Op1`	`Read shape`	• read and comprehend the overall shape of a star plot
`Op2`	`Compare two shapes`	• compare two shapes and decide whether they are highly similar
`Op3`	`Read scale value`	• read and comprehend a scale value
`Op4`	`Find correspond-ing scale`	• after having read a scale name or value, find the corresponding scale in another picture
`Op5`	`Compare two scale values`	• compare two scale values and decide whether they are identical
`Op6`	`Read scale name`	• read and comprehend the scale name

Task 1: **Highly Similar** *The Algorithm*		`For TWO star plots`
		`Read shape A (Op1)`
		`Read shape B (Op1)`
		`Compare two shapes A + B (Op2)`

The complexity		`2*Op1 + Op2`

Task 2: **Identify Differences** *The Algorithm*		`For TWO star plots for EACH Scale`
		`Read scale value (Op3)`
		`Find corresponding scale (Op4)`
		`Read scale value B (Op3)`
		`Compare two scale values (Op5)`
		`Read scale name (Op6)`

The complexity		`#C * (2*Op3 + Op4 + Op5 + Op6)`

Table 2. Example 1: Tasting Whisky, defining the elements needed to proceed with the complexity analysis

The result of the task 1, which checks for highly similar features of whiskys is `2*Op1+Op2` and the result of the task 2, which identifies the main differences of the whiskys is `#C * (2*Op3 + Op4 + Op5 + Op6).`

The complexity metric *ViCo* could be easily expanded to consider further issues (e.g., `Var2=#W`, which covers the number of whiskys to be compared) or it could be refined (e.g., `Var3= #Identical` counts the scales that do not show substantial differences).

Looking at an additional visualization that shows differences of corresponding features (see Figure 2), the complexity and the savings can be easily computed. Simplifying, if we assume that no additional operation is needed for finding the next scale with a difference (which only holds true for small numbers) and *`#C`* is the number of spokes and *`#Identical`* is the number of identical strokes, then the complexity function would be `(#C - #Identical) * (Op3+Op6).`

If we compare this result with the result of Table 2 (Task 2), them we can easily recognize that the computational complexity of the second visualization applying the same operators is much easier than the complexity of the other visualization.

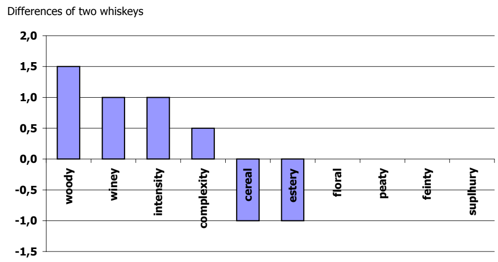
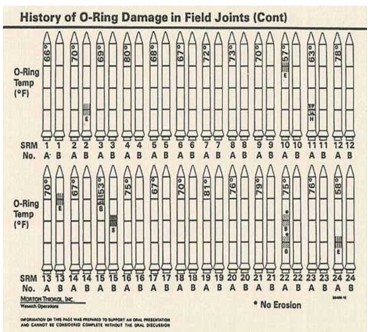


Fig. 2. Visualizing the differences of the two whiskys of Figure 1.

Further possible expansions of the analysis might consider, for example, how users would deal with large numbers of comparisons. Then also user interactions (e.g., selecting two whiskys and switching to another visual representation) might become relevant basic operations. A closer look at the basic operations might also lead to refinement of the metric and to a better understanding of the visualization. E.g., to what number of scales is a reading and comparison of shapes as a single operation reasonably possible? To what precision is the reading of scale values possible?

4.2 Example 2: Visualizations of some Issues Regarding the Challenger's Disaster

Within the field of Information Visualization, scatter plots are another important class of diagrams and visual aid. Such diagrams can lead to great insight, but also to its occlusion. As an example for this Tufte [18] cites the accident of the space shuttle Challenger.



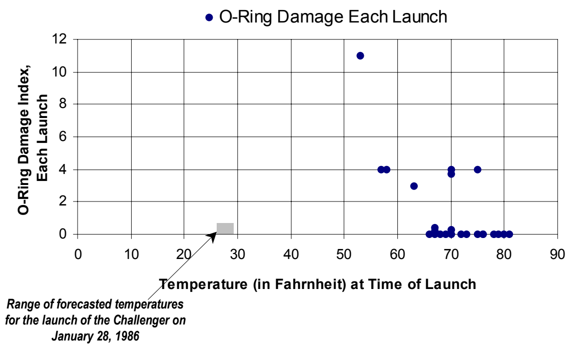


Fig. 3. Visualization of the shuttle's disaster showing the original diagram used by the NASA and the booster rocket manufacturer.

Fig. 4. Visualization of the shuttle's disaster showing the final re-visualization by Tufte [18].

in: Second International Conference on Theory and Application of Diagrams (Diagrams 2002), Springer, Berlin, 2002.

There was the question about whether the shuttle should be launched on a cold day (January 27, 1986). The decision depended on whether the temperature would make the O-Ring that sealed the sections of the booster rocket unsafe.

The Figure 3 reprints one of the diagrams used in making the decision by the booster rocket manufacturer. Based on that diagram, NASA decided to launch the shuttle and the O-Ring was damaged and the shuttle crashed. The next Figure 4 shows the re-visualization by Tufte [18]. It uses a simple scatter plot depicting the relation between the two major variables of interest. Different types of damage are combined into a single index of severity. The proposed launch interval of temperature is also put on the chart to show it in relation to the data. The new diagram tentatively indicates a pattern of damage below 70^{o} or 60^{o}.

In the following we discuss these two visualizations in terms of *ViCo* for the same task of understanding whether there is a relation between temperature and O-Ring damage. Table 3 shows the use of *ViCo* in detail.

Kind	**Name**	**Explanations**
`Vis2`	`Diagrams`	See above Figure 3 and 4
`T`	`Relate Damage and O-Ring`	Is there a relation between O-Ring damage and temperature?
`Users`		• are able to understand and read scatter plots
`Var`	`#N` `#O`	• number of shuttle's starts • number of O-Rings
`Op1`	`Read Damage`	• read and comprehend severity of O-Ring damage
`Op2`	`Read Temperature`	• read and comprehend the temperature
`Op3`	`Write Data`	• write down in corresponding column the damages of O-Rings and temperatures
`Op4`	`Calculate Measures`	• compute the average and the measures needed to compare the data series
`Op5`	`Read Shape & Decide`	• read and comprehend the overall shape/distribution of the data points • make decision whether there is a very clear relationship
`Op6`	`Read Data Points`	• read and comprehend data points

Task: **Relate Damage and O-Ring with Fig. 3**
The Algorithm
		`For EACH start (#N)`
		`  For EACH O-Ring (#O)`
		`    Read Damages (Op1)`
		`  Read Temperature (Op2)`
		`  Write Data (Op3)`
		`  Calculate Measures (Op4)`

The complexity of Fig 3.		`#N * (#O * Op1 + Op2 + Op3 + Op4)`

Task: **Relate Damage and O-Ring with Fig. 4**
The Algorithm
	`Read Shape & Decide(Op5)`
	`IF no clear Relation THEN`
	`   For EACH Data-Point (#N)`
	`      Read Data Points (Op6)`
	`      Calculate Measures (Op4)`

The complexity of Fig. 4	`Best Case: Op5` `Worst Case: Op5 + #N*(Op6 + Op4)`

Table 3. Example 2: Tufte's visualization of some issues regarding the Challenger Disaster, defining the elements needed to proceed with the complexity analysis

In the case of Figure 3, to achieve that task, 'normal' users will need a lot of processing (including ordering and calculating appropriate measures, e.g., calculating the averages of damage/no damage launches). In the case of Figure 4, users can start with capturing the shape of the data series, because the scatter plot is already structured according to the two variables of interest. If a very clear picture emerges, the task is achieved. Otherwise, again calculation is necessary. In Section 2 in the paragraph about the complexity, we mentioned that complexity analysis in computer science often provides both *upper* and *lower bounds*. In the above example for the complexity measure of Figure 4, best case and worst case are these bounds.

In order to facilitate comparisons of the complexity measures of Figure 3 and Figure 4, we further simplify their measures. As there are several operations, each with constant length, we can introduce scalars (`a, b, c,` etc.) to express one as a multiple of the other (compare Section 3.3). Let `Op1` take `X` seconds, then `Op2` takes `a*X` seconds, `Op3` takes `b*X` seconds, etc. Transforming the complexity functions from Table 3, we get the following new measures

The complexity of Fig. 3.	`#N*(#O + a + b + c)*X`
The complexity of Fig. 4.	`Best Case: d*X` `Worst Case: (d + #N*(e + c))*X`

With further simplifications we can compare the complexity of Figure 3 with the worst case complexity of Figure 4: $\frac{\#O + a + b + c}{\frac{d}{\#N} + e + c}$. With large `#N` we arrive at

$\frac{\#O + a + b + c}{e + c}$ which shows that the visualization in Figure 3 is much worse as long as `Op6` is not more complex than `(#O+a+b)*X`.

The comparison of the complexities still includes the element c (coming from `Op4` – calculating measures). This indicates that – if there is no clear shape – computation is still necessary. For the example given, the data point at 53° strongly shapes the overall impression. If this point would be considered to be an outlier then the picture would be less clear and correspondingly computation necessary. It might be the case that classical statistics is better and more informative to apply then. Still, the visualization of Figure 4 would allow for an extremely quick check whether there is a very clear relationship or whether calculation is necessary.

5 Discussion and Conclusions

In the paper presented, we have argued for *ViCo*, a metric for the complexity of various diagrams or more general approaches dealing with Information Visualization. For this purpose, we defined several conceptual fundamentals: Tasks and Users, Reading and Writing, Comparing and Calculation, and Complexity. Our approach is mainly influenced and guided by two scientific fields, on the one side, the algorithmic thinking and complexity theory in computer science [13] and, on the other side, the study of cognition and perception in psychology [7].

Our goals were to utilize concepts from perception and cognition to arrive at measures to judge the readability and the complexity of visualizations. We are definitely aware that perception and cognition work differently than algorithmic thinking (for example, we did not address, how we are dealing with know-how or any kind of learning effects to ease and facilitate the understanding of diagrams). We have knowingly simplified some cognitive aspects (e.g., memorization of information, know-how, learning) because we argue that in spite of such simplifications meaningful comparisons can be made. Similar considerations hold true for temporal aspects. Sometimes it may be necessary to actually measure times (e.g., with eye tracking). However, in many cases *ViCo* can work without such measurement.

We are not aiming to explain intuitive understanding of diagrams or any kind of visualizations. Additionally, we do not compare oranges with apples or scatter plots with danger signs. *ViCo* goes for a smaller but still reasonable aim. We analyze the readability of diagrams with respect to particular users and tasks. This means we are comparing oranges of kind A with oranges of kind B.

Finally, we aim 'only' for *relative complexity* of visualizations and not for *absolute complexity*. I.e., we do not attempt to develop a metric that covers all possible visualizations, for all possible tasks for all possible user groups. We go for smaller objective: We want to be able to compute the complexity for any given set of visualizations with given tasks and given assumptions regarding the user group allowing for incremental enlargement for any specific field.

ViCo does (to some degree) analogous things in the field of Information Visualization as GOMS does in the field of user interface design. GOMS (Goals, Operators, Methods, and Selection rules) [9] is an analytical analysis technique. The goal of GOMS is to radically reduce the time and cost of designing usable systems through developing analytic engineering models for usability tests based on validated computational models of human cognition and performance. The GOMS family provides various methods to count and measure how long a user needs to accomplish a task using a particular tool. Many variants of GOMS rely on measuring and calculating actual times, which limits the field of application and makes it more difficult to apply it for new types of information processing. However, this is just what Information Visualization aims for. Furthermore, some limitations of GOMS are inherited in our approach *ViCo* too, like differences between users, learning process, mistakes in executing the basic operations and inside the interpretation step.

We are well aware of the fact that the procedures described above touch a high number of questions that cannot be solved in general (e.g., comparison of algorithms, accelerate possible algorithm, definition of minimal operations). However, these questions can be tackled to an acceptable degree in most practical situations.

Correspondingly, the procedure can contribute to better-informed decision making on which visualization to use when in a way that is not possible with direct observation or by measuring only the time that is needed to accomplish tasks as a whole.

Acknowledgements. We thank Monika Lanzenberger, Peter Purgathofer, Robert Spence, Jessica Kindred, and the anonymous referees for their useful and constructive comments. The Asgaard Project is supported by „Fonds zur Förderung der wissenschaftlichen Forschung“ (Austrian Science Fund), grant P12797-INF.

References

1. Baase, S. & Van Gelder, A.: *Computer Algorithms: Introduction to Design and Analysis* 3rd ed.). Addison Wesley, Reading, Mass. [u.a.], (2000).
2. Berg, M.: Accumulating and Coordinating: Occasions for Information Technologies in Medical Work. *Computer Supported Cooperative Work: The Journal of Collaborative Computing,* 8 (1999) 373-401.
3. Card, S. K., Mackinlay, J. & Shneiderman, B. (Eds.): *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco (1999).
4. Chambers, J., Cleveland, W., Kleiner, B. & Tukey, P.: *Graphical Methods for Data Analysis,* Wadsworth, (1983).
5. Gärtner, J.: *Software in Consulting.* Habilitationsschrift Thesis, Technische Universität Wien, (2001).
6. Gärtner, J. & Wahl, S.: The Significance of Rota Representation in the Design of Rotas. *Scandinavian Journal of Work, Environment & Health,* 24(3) (1998) 96-102.
7. Goldstein, E. B.: *Sensation and Perception* 5th ed.). Brooks/Cole Publishing Company, (1998).
8. Grinstein, G. G., Hoffman, P. E. & Pickett, R. M.: Benchmark Development for the Evaluation of Visualization for Data Mining. In Fayyad, U., Grinstein, G. G., et al. (eds.), *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann, San Francisco, (2002) 129-176.
9. John, B. E. & Kieras, D. E.: The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast. *ACM Transactions on Computer-Human Interaction,* 3 (1996) 320-351.
10. Kosara, R. & Miksch, S.: Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue,* 22(2) (2001) 111-131.
11. MacEachren, A. M.: *How Maps Work* The Guilford Press, New York, (1995).
12. Miksch, S., Horn, W., Popow, C. & Paky, F.: Utilizing Temporal Data Abstraction for Data Validation and Therapy Planning for Artificially Ventilated Newborn Infants. *Artificial Intelligence in Medicine,* 8(6) (1996) 543-576.
13. Papadimitriou, C. H.: *Computational Complexity* Addison Wesley, Reading, Mass. [u.a.], (1994).
14. Schumann, H. & Müller, W.: *Visualisierung: Grundlagen und allgemeine Methoden* Springer, Berlin, (2000).
15. Spence, R.: *Information Visualization* ACM Press, New York, (2001).
16. Tufte, E. R.: *The Visual Display of Quantitative Information* Graphics Press, Cheshire, CT, (1983).
17. Tufte, E. R.: *Envisioning Information* Graphics Press, Cheshire, CT, (1990).
18. Tufte, E. R.: *Visual Explanation* Graphics Press, Cheshire, CT, (1997).