EuroVis Workshop on Visual Analytics (2019)
T. von Landesberger and C. Turkay (Editors)

On Quality Indicators for Progressive Visual Analytics

M. Angelini[1], T. May[2], G. Santucci[1], H.-J. Schulz[3]

[1]Sapienza University of Rome, Italy
[2]Fraunhofer Institute for Computer Graphics Research, Germany
[3]Aarhus University, Denmark

Abstract

A key component in using Progressive Visual Analytics (PVA) is to be able to gauge the quality of intermediate analysis outcomes. This is necessary in order to decide whether a current partial outcome is already good enough to cut a long-running computation short and to proceed. To aid in this process, we propose ten fundamental quality indicators that can be computed and displayed to gain a better understanding of the progress of the progression and of the stability and certainty of an intermediate outcome. We further highlight the use of these fundamental indicators to derive other quality indicators, and we show how to apply the indicators in two use cases.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics;*** • ***Computing methodologies*** → ***Progressive computation;***

1. Introduction

In times of large data volumes and sophisticated long-running computational analyses, Progressive Visual Analytics (PVA) is rapidly becoming the new data analysis approach of choice. Unlike most existing analysis approaches, PVA does not compute the whole dataset at once, but either processes large data volumes in chunks or long-running computations in steps [ASSS18]. This way, intermediate outcomes† can be shown to the analyst while the computation is still running. Unlike stream processing, progressive computations are bounded – i.e., a final outcome will be obtained at some point. Until then, a good-enough intermediate outcome can be used instead of the final one to inform early decisions or to jump start subsequent analysis steps. Yet this requires the analysts to gauge when to stop the running computation based on only those data and iterations they saw so far. The challenge they face is how to assess the quality of an intermediate PVA outcome in relation to a still unknown final one?

Data quality has long been an active field of research in visualization [JF17, BAOL12] and visual analytics [LAW*18, SSK*16]. Typically, these notions relate to flawed input data – e.g., missing values, duplicate entries, or uncertain measurements. Whereas the quality of visualizations usually aims to measure the perceptual goodness (or badness) of the view – e.g., the amount of overplotting or the number of edge crossings [BBK*18].

† Note that we use the term *outcome* whenever a statement holds for both, numerical results from a progressive computation and views generated by a progressive visualization. Otherwise, we clearly denote the respective outcome as a *result* or as a *view*.

In PVA however, the idea of quality is a slightly different one, as any lack thereof is typically thought as having been introduced by the progression itself and the fact that it conveys only an incomplete subset of the whole data – i.e., the longer one waits, the more data or iterations could be processed, and the higher the quality of the outcome. This leads to a very process-oriented notion of quality that aims to appraise an intermediate outcome with respect to the running progression.

To better judge outcomes, the literature recognizes the importance of communicating information about the running process [VCR16]. For progressive processes, mainly *progress* and the *uncertainty* estimations are used [MPG*14, BEF17, TKBH17]. In other cases, the complementary notions of *fluctuation* [TKBH17] and *stability* [FFK14, vLAA*13] are taken into account as well.

Only recently have these individual approaches been discussed and put into the general context of a "stack" that bundles these individual quality aspects and gives first informal definitions for them [FFNE18]. This work forms the outset of this paper, in which we further explore the idea of providing a more nuanced perspective on quality in PVA by making the following contributions:

- differentiating between quality of the input, of the numerical result, and of the generated view;
- differentiating between absolute quality and relative quality;
- enumerating the fundamental quality indicators resulting from those differentiations; and
- introducing the notion of composite quality indicators.

We further illustrate the use of some of the introduced quality indicators in two use cases.

DOI: 10.2312/eurova.20191120

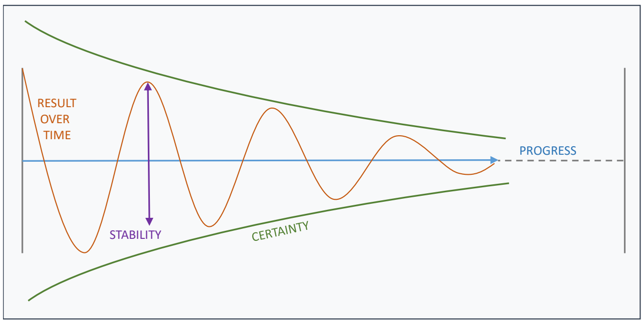

Figure 1: *A prototypical result of a progressive computation and its principal quality properties: its certainty (e.g., given as confidence interval), its stability (e.g., given as a convergence measure), and its progress (e.g., given as the percentage of data already processed).*

2. Quality Indicators in PVA

Along the lines of the mentioned stack of quality measures [FFNE18], this section discerns between three fundamental notions of quality in PVA: the *progress* of the progressive process, the *stability* of this process, and the *certainty* of this process' outcomes. A visual overview of the traits of the progression that these three quality indicators describe and how they relate to each other can be seen in Figure 1. Based on them, we now further differentiate between the domain in which the quality is measured, as well as whether this measurement is an absolute or a relative one. We then detail their use as building blocks for composite quality indicators.

2.1. Quality Indicators for Different Domains

PVA incorporates by design progressive computational analysis and progressive visualization. In its simplest form, the PVA process takes some data as input, computes a numerical result, which is then visualized resulting in a view – only then to start with the next increment of data or with the next iteration of the computation. This results in three entities for which we can measure quality:

The input data, in cases where it is already progressively provided – e.g., when data samples are progressively funneled into the PVA pipeline. Not only can we measure basic properties like progress directly on the input data (e.g., how much data out of all data has already been processed), but also characteristics of the data sampling (e.g., whether it is truthful to a known or assumed distribution).

The results of the computation that are progressively improved by taking more data into account, or refined by more and more iterations. For them, we could measure their stability by quantifying the changes they undergo from one intermediate result to the next. Another option to establish a notion of result stability is to have two PVA processes run in parallel, which work on different data samples and whose convergence or divergence can be measured.

The view being refined by the progressive visualization that generates an increasingly complete view by incorporating more and more computational results, or that iteratively refines the view. For the view, we can likewise determine stability, as for example the underlying computation results may still be further refining, but not lead to any more visible changes.

2.2. Absolute and Relative Quality Indicators

Depending on whether we aim to assess the quality of an intermediate outcome in relation to the anticipated final outcome, or in relation to prior outcomes, we further discern between:

Absolute quality indicators quantifying the quality with respect to a known or estimated final, and thus best possible and most accurate outcome. This notion captures how much of the final outcome has already been achieved – for example, in terms of data processed or in terms of error still inherent in the intermediate outcome.

Relative quality indicators quantifying the quality with respect to a prior state. These measures capture if an outcome has improved or not as compared to a prior state, and by how much it has done so. If improvement cannot be discerned, at least change (for better or worse) can be detected and gauged.

2.3. Fundamental Quality Indicators

Applying the above distinctions between input, result, and view as well as between absolute and relative indicators, we can establish ten fundamental notions of PVA quality, which are listed in Table 1.

2.3.1. Progress Indicators

Progress is defined as the amount of advancement achieved by the progression at a point in time – e.g., the number of data items already processed, the number of iterations already completed, or simply the time elapsed. While its usefulness for gauging the quality of an intermediate outcome is limited [FFNE18], the average user is certainly more versed in interpreting progress bars than error bars, which may be why progress is frequently indicated in PVA systems. We discern mainly between absolute and relative progress.

Absolute progress $AP_i \in [0 \ldots 1]$ at outcome i can be computed as the proportion $AP_i = doneWork_i / totalNeededWork$. In many cases (estimated) information about the $totalNeededWork$ is available: For iterative processes, we can utilize known average or worst case complexities for estimating a needed overall number of iterations. Similar estimates are available for many other iterative algorithms – e.g., k-means [AMR11]. For the incremental processing of data chunks, we can use the data size to estimate the needed work. Moreover, it is possible to define AP in terms of all domains:

- $AP_{input_i} = processedData_i / sizeOfData$.
- $AP_{result_i} = sizeOfResult_i / expectedResultSize$. This is what we call the *Computational Yield*, which is for example the current number of search results as compared to an estimate – e.g., searching for primes in a set of numbers or for motifs in a graph.
- $AP_{view_i} = renderedElements_i / numberOfElements$. Using Tufte's concept of *ink* to denote non-background pixels [Tuf01], we can measure the progress of background pixels turned into foreground pixels – e.g., in scatterplots – while disregarding any overplotting of pixels already colored.

Relative progress RP is available as soon as at least two consecutive progression steps $i-1$, i are available. It is computed by means of the absolute progress indicators: $RP_{x_i} = AP_{x_i} - AP_{x_{i-1}}$, in range $[0 \ldots 1]$, with $x \in \{input, result, view\}$. Note that depending on the used computation and visualization, a large RP_{input} does not necessarily produce a large RS_{result}, which in turn does not necessarily lead to a large RS_{view}.

Type	Domain	Symbol	Example
Absolute Progress	Input	AP_{input}	Processed Data Items, Completed Iterations
	Result	AP_{result}	Computational Yield – e.g., found search results
	View	AP_{view}	Deposited Ink – e.g., colored pixels in a scatterplot
Relative Progress	Input	RP_{input}	Processed Data per Iteration
	Result	RP_{result}	Computational Yield per Iteration
	View	RP_{view}	Ink Deposited per Iteration
Relative Stability	Input	RS_{input}	Change in Value Distribution between Data Chunks
	Result	RS_{result}	Change in Numeric Output per Iteration
	View	RS_{view}	Change in Visual Output per Iteration
Absolute Certainty	Result	AC	Confidence Interval

Table 1: *List of the ten proposed fundamental quality indicators.*

2.3.2. Stability Indicators

Stability is defined as the amount of change, deviation, or fluctuation the progression exhibits, as compared to a sequence of outputs that monotonously converges. Since it captures a property of the process of the progression, there exists no absolute stability for individual outcomes. **Relative stability** *RS* can be computed once at least two progression steps $i-1$, i are available and it can be derived for input, result, and view as $\Delta_i/status_{i-1}$, i.e., the ratio between the variation at i by the status at $i-1$, ranging in $[0\ldots+\infty]$:

- RS_{input_i}, e.g., as the difference between the means of two consecutive inputs $|Mean_i - Mean_{i-1}|/Mean_{i-1}$;
- RS_{result_i} between two partial results, e.g., $1/Jaccard(result_i, result_{i-1})$ of a PVA progressively computing a set with specific properties;
- RS_{view_i}, e.g., the ratio $\sum_j |barHeight_{j,i} - barHeight_{j,i-1}|/barHeight_{i-1}$ of the heights of the bars of a barchart (see, e.g., [AS17]), or the ratio [dots in a scatterplot that changed color] / [number of data elements].

2.3.3. Certainty Indicators

Certainty (which [FFNE18] calls *quality*) is defined as the amount of error by which the actual, final result could still deviate from the current result. In this case, only the **absolute certainty** *AC* makes sense, i.e., the certainty associated with a current partial result. *AC* is considered a measure of the computational result, but it is actually derived from the input – e.g., the closer the incoming data matches the overall distribution of the dataset, the more certain are the results computed from them. If statistical information about the dataset as a whole is available, we can use it for expressing certainty, e.g., providing confidence intervals. If such information is not available, we can investigate the data distribution properties (e.g., through density estimation) considering the incoming data as sampling without replacement or to infer other properties, e.g., estimating the maximum through frequentist or Bayesian inference.

2.4. Composite Quality Indicators

Fundamental quality indicators can be used as is in various scenarios. For example, RP_{view} can be a key indicator to decide whether to present the user with a new outcome – i.e., it makes no sense to update a view if the changes will be barely visible. Furthermore, these indicators can be combined for forming substitutes for missing indicators or for deriving entirely new indicators. In what follows, we focus on some meaningful examples of such combinations.

2.4.1. Deriving Substitute Indicators

While some indicators may be readily available, it can be hard to measure others. In the simplest case, one type of indicators can simply stand in for another, providing a rough estimation – e.g., $AC \approx AP$. Yet it is also possible to bring multiple indicators into the picture and to combine them through a linear combination: $AC \approx (\alpha * AP + \beta * RS_{result} + \gamma * RS_{view})$. It is worth noting that in this case we have to normalize RS in the range $[0\ldots1]$ (as an example, we can compute the RS_{input_i} as $|Mean_i - Mean_{i-1}|/(Mean_{i-1} + Mean_{i-1})$ that ranges in $[0\ldots1]$).

2.4.2. Deriving New Indicators

The fundamental indicators can also be used to derive entirely new ones that carry extra meaning and help to discern quality aspects that can only be found in these combinations.

Expressiveness means that the underlying data is truthfully represented in a visualization. This concept harkens back to Tufte's idea of a *lie factor* [Tuf86] and has been picked up as *visual-data correspondence* [KS14] or *preservation task* [BBK*18] in the literature. In PVA, expressiveness is equated to *change proportionality* [ASSS18] – i.e., the observable visual change between two intermediate views being proportional to the change between the underlying results: $Exp = RP_{view}/RP_{result}$. A high *Exp*-value signals an overemphasis of the changes between results in the view, whereas a low *Exp*-value signals that the view downplays the changes.

Certainty variation captures the relationship between the change in certainty of the current partial result and the computational progress with respect to the previous one: $AC_{var} = \Delta AC/\Delta AP_{input}$, where AP_{input} can be measured in data chunks or iterations. An example of its application are results from progressive *t*-SNE [PLvdM*17], where each iteration brings quality improvements, eventually decreasing when the algorithm gets stable.

Progression trustability expresses the overall trustability as a (weighted) combination of Stability, Certainty, and Progress: $Trust = f(RS, AC, AP)$. An example of low progression trustability is a progressive Treemap rendering [RH09], that exhibiting layout changes even with high *AP* and *AC* values. This behavior can lead the user to have less trust in the underlying partial results.

3. Use Case

Here we give two examples illustrating when and how the indicators can be used to properly gauge quality. In both examples, we assume that the data to be processed is fixed from the start.

3.1. Probability Mass Function Estimation

In our first example, we discuss the estimation of a *probability mass function* (PMF), which represents a density distribution of a discrete univariate attribute. The PMF can be used, e.g., for determining distribution classes, multivariate correlation analysis, or

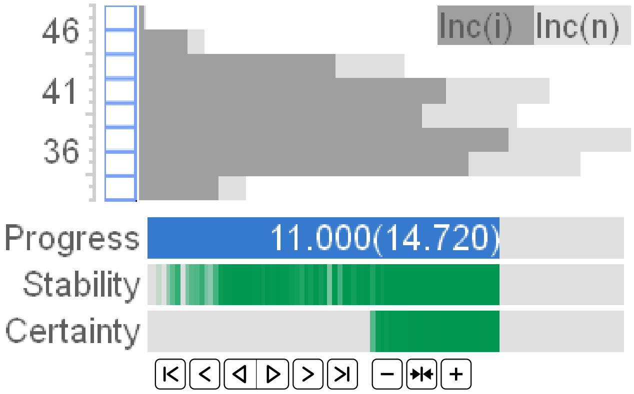

Figure 2: *This example illustrates the first use case. The histogram shows the distributions Inc_i and Inc_n of human body measures (shoulder width). The progress bars for stability and certainty (green) map the indicator values to color saturation. The data sample is generated from two copies of the same data pool. The copy that is loaded with the first chunks has been sorted, which introduces a strong sampling bias, resulting in low stability and certainty. The second copy contains randomly ordered data, causing a prompt increase of the indicators. (Note: To play the animation, a standalone PDF viewer is required)*

histogram visualizations (see Figure 2). The process runs incrementally using n chunks of data, assuming m attribute values or bins. A sample increment Inc_i aggregates the values of all chunks $1 \dots i$.

$$Inc_i = \left(c_{i,1}, c_{i,2}, \dots, c_{i,m}\right) \in \mathbb{N}^m \qquad (1)$$

Useful quality indicators can now be described as follows.

Progress: A simple measure of last resort, $AP := i/n$.

Stability: Stability represents differences between one (possibly more) consecutive increments Inc_{i-1} and Inc_i. As the former can be considered a sample drawn from the latter, we suggest using the *Chi-Square-Goodness-Of-Fit* statistic.

$$Chi^2\left(Inc_i, Inc_{i-1}\right) = \sum_{j=1}^{m} \frac{\left(c_{i,j} - c_{i-1,j}\right)^2}{c_{i,j}} \qquad (2)$$

It accounts for the sample size and is very sensitive to sampling biases. The stability metric is the p-value derived from the statistic.

Certainty: Measuring certainty requires the PMF of the entire data Inc_n being known *from the start*. This is realistic, as a PMF is not always the actual result of a process, but a proxy for its input. Consider, for example, a progressive training of a classifier. A PMF can be calculated for every training batch (=sample increment) and remains meaningful even if the total distribution is known in advance. The certainty reflects that the increment Inc_i is in fact a representative of the entire data, which in turn is a prerequisite for the validity of the classifier. Again, we use the Chi-Square measure here. Certainty is measured as the p-value of $Chi^2\left(Inc_n, Inc_i\right)$.

Note that the chosen metrics may not be the best choice in other cases. For example, our stability metrics focus on the sample increments, instead of the difference between two increments (i.e., a chunk). If this stability of the difference is more important in subsequent calculations, this needs to be considered accordingly.

3.2. *k*-Means

In our second example, we discuss the estimation of the quality indicators of a *k-Means* clustering. In this case, the calculation is run iteratively, with the entire dataset being available from the start. In this example $i : 1 \dots n$ denotes the number of the current iteration, with n assumed to be a maximum number of iterations.

Progress: Similar to the first example, but counting iterations instead of chunks, $AP := i/n$.

Stability: With iterative processes operating on fixed (i.e. 'stable') data, stability is measured between iterations. Equivalent choices are the cluster differences and the movement of the centroids.

Certainty: Here, we cannot assume that the optimal clustering result is known in advance. Even when the method has converged, this is no indication that a globally optimal solution has been found. Thus, certainty cannot be estimated from a reference solution. Yet it is possible to estimate the maximum of an unknown set of numbers from a random sample of this set, as long as its distribution is known – cf. *German Tank Problem* [RB47]. In our use case this 'set of numbers' are all quality measures derived from the clustering results of all iterations. While many evaluations have been made with optimal solutions, few, if any, studies analyze the quality distribution of intermediate solutions. As of yet, this approach remains to be validated to complete the set of measures for iterative problems.

4. Conclusion

In this paper, we introduced two important differentiations on top of the existing quality concepts in PVA: the distinction between absolute and relative quality, as well as the distinction between quality of input, result, and view. From those, we derived ten fundamental quality indicators that can be combined into other meaningful indicators. From our own experience of working with them in the use cases, we can conclude the following:

- Results of high certainty can be used as final results, and the computation can be halted early.
- Results of high stability can already be used for early visualization and interactive exploration.
- Low quality values until the end signify a skewed sampling, which is usually introduced by ordered data.

In particular the last case is a challenge, as many datasets come with an inherent order (e.g., time-varying data) or the indexing structures of the database keep and return the data in such order. While the proposed quality indicators can already help to identify these cases, it remains a question for future research on how to effectively counter these effects – in particular in those cases where random shuffling is not applicable.

Acknowledgements

We are indebted to the organizers and participants of Dagstuhl Seminar 18411 "Progressive Data Analysis and Visualization" whose discussions on uncertainty in PVA posed the outset for this work. We also thank the anonymous reviewers for their comments.

References

[AMR11] ARTHUR D., MANTHEY B., RÖGLIN H.: Smoothed analysis of the k-means method. *Journal of the ACM 58*, 5 (2011), 19:1–19:31. doi:10.1145/2027216.2027217. 2

[AS17] ANGELINI M., SANTUCCI G.: On visual stability and visual consistency for progressive visual analytics. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (IVAPP)* (2017), Linsen L., Telea A., Braz J., (Eds.), SciTePress, pp. 335–341. doi:10.5220/0006269703350341. 3

[ASSS18] ANGELINI M., SANTUCCI G., SCHUMANN H., SCHULZ H.-J.: A review and characterization of progressive visual analytics. *Informatics 5*, 3 (2018), 31. doi:10.3390/informatics5030031. 1, 3

[BAOL12] BRODLIE K., ALLENDES OSORIO R., LOPES A.: A review of uncertainty in data visualization. In *Expanding the Frontiers of Visual Analytics and Visualization*, Dill J., Earnshaw R., Kasik D., Vince J., Wong P. C., (Eds.). Springer, 2012, pp. 81–109. doi:10.1007/978-1-4471-2804-5_6. 1

[BBK*18] BEHRISCH M., BLUMENSCHEIN M., KIM N. W., SHAO L., EL-ASSADY M., FUCHS J., SEEBACHER D., DIEHL A., BRANDES U., PFISTER H., SCHRECK T., WEISKOPF D., KEIM D. A.: Quality metrics for information visualization. *Computer Graphics Forum 37*, 3 (2018), 625–662. doi:10.1111/cgf.13446. 1, 3

[BEF17] BADAM S. K., ELMQVIST N., FEKETE J.-D.: Steering the craft: UI elements and visualizations for supporting progressive visual analytics. *Computer Graphics Forum 36*, 3 (2017), 491–502. doi:10.1111/cgf.13205. 1

[FFK14] FERREIRA N., FISHER D., KÖNIG A. C.: Sample-oriented task-driven visualizations: Allowing users to make better, more confident decisions. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)* (2014), ACM, pp. 571–580. doi:10.1145/2556288.2557131. 1

[FFNE18] FEKETE J.-D., FISHER D., NANDI A., (EDS.) M. S.: Progressive data analysis and visualization. *Dagstuhl Reports 8*, 10 (2018), 1–40. doi:10.4230/DagRep.8.10.1. 1, 2, 3

[JF17] JOSKO J. M. B., FERREIRA J. E.: Visualization properties for data quality visual assessment: An exploratory case study. *Information Visualization 16*, 2 (2017), 93–112. doi:10.1177/1473871616629516. 1

[KS14] KINDLMANN G., SCHEIDEGGER C.: An algebraic process for visualization design. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2181–2190. doi:10.1109/TVCG.2014.2346325. 3

[LAW*18] LIU S., ANDRIENKO G., WU Y., CAO N., JIANG L., SHI C., WANG Y.-S., HONG S.: Steering data quality with visual analytics: The complexity challenge. *Visual Informatics 2*, 4 (2018), 191–197. doi:10.1016/j.visinf.2018.12.001. 1

[MPG*14] MÜHLBACHER T., PIRINGER H., GRATZL S., SEDLMAIR M., STREIT M.: Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1643–1652. doi:10.1109/TVCG.2014.2346578. 1

[PLvdM*17] PEZZOTTI N., LELIEVELDT B. P. F., V. D. MAATEN L., HÃŰLLT T., EISEMANN E., VILANOVA A.: Approximated and user steerable t-SNE for progressive visual analytics. *IEEE Transactions on Visualization and Computer Graphics 23*, 7 (2017), 1739–1752. doi:10.1109/TVCG.2016.2570755. 3

[RB47] RUGGLES R., BRODIE H.: An empirical approach to economic intelligence in World War II. *Journal of the American Statistical Association 42*, 237 (1947), 72–91. doi:10.1080/01621459.1947.10501915. 4

[RH09] ROSENBAUM R., HAMANN B.: Progressive presentation of large hierarchies using Treemaps. In *Advances in Visual Computing* (2009), Bebis G., Boyle R., Parvin B., Koracin D., Kuno Y., Wang J., Pajarola R., Lindstrom P., Hinkenjann A., Encarnação M. L., Silva C. T., Coming D., (Eds.), Springer, pp. 71–80. doi:10.1007/978-3-642-10520-3_7. 3

[SSK*16] SACHA D., SENARATNE H., KWON B. C., ELLIS G., KEIM D. A.: The role of uncertainty, awareness, and trust in visual analytics. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 240–249. doi:10.1109/TVCG.2015.2467591. 1

[TKBH17] TURKAY C., KAYA E., BALCISOY S., HAUSER H.: Designing progressive and interactive analytics processes for high-dimensional data analysis. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 131–140. doi:10.1109/TVCG.2016.2598470. 1

[Tuf86] TUFTE E. R.: *The Visual Display of Quantitative Information*. Graphics Press, 1986. 3

[Tuf01] TUFTE E. R.: *The Visual Display of Quantitative Information*, 2nd ed. Graphics Press, 2001. 2

[VCR16] VERAME J. K. M., COSTANZA E., RAMCHURN S. D.: The effect of displaying system confidence information on the usage of autonomous systems for non-specialist applications: A lab study. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)* (2016), ACM, pp. 4908–4920. doi:10.1145/2858036.2858369. 1

[vLAA*13] VON LANDESBERGER T., ANDRIENKO G., ANDRIENKO N., BREMM S., KIRSCHNER M., WESARG S., KUIJPER A.: Opening up the "black box" of medical image segmentation with statistical shape models. *The Visual Computer 29*, 9 (2013), 893–905. doi:10.1007/s00371-013-0852-y. 1