SIGRAD 2012
A. Kerren and S. Seipel (Editors)

# Visual Parameter Optimization for Biomedical Image Analysis: A Case Study

A.J. Pretorius[1], D. Magee[1], D. Treanor[2,3], and R.A. Ruddle[1]

[1]School of Computing, University of Leeds
[2]Leeds Teaching Hospitals Trust
[3]Leeds Institute of Molecular Medicine, University of Leeds

---

**Abstract**

*The conventional approach for parameter optimization of biomedical image analysis algorithms is to tweak parameters by trial-and-error. This presents a challenge: parameter space is often inadequately explored and, consequently, output quality suffers. Interactive visualization can alleviate this problem but has not been widely adopted. Moreover, few examples of the successful application of visualization for parameter optimization of image analysis algorithms have been published. To address this and to illustrate the potential usefulness of interactive visualization, we present a case study. A multidisciplinary team developing novel image segmentation software for histopathology was observed. Within the context of our study, our hypotheses were confirmed: (1) using interactive visualisation, participants considered larger parts of parameter space than they had previously by trial-and-error; (2) participants gained a better understanding of their algorithm (an unknown logic error and errors in its implementation were discovered); and (3) participants achieved higher quality output. Our work is also an example of the value of case studies in iterative design. We describe how a valuable additional requirement was revealed (the importance of derived measures) and how our visualization method was extended to cater for this.*

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Visualization; I.3.8 [Computer Graphics]: Applications—Biomedicine

---

## 1. Introduction

Parameter optimization is often encountered in applied science and engineering and presents a non-trivial challenge: given a system with several input parameters that produces one or more outputs, how do users find suitable parameter values such that certain requirements hold for the outputs? In particular, the application of segmentation algorithms to biomedical images requires users to find parameter values to accurately detect objects, such as cells, or higher-level structures, such as regions of tissue. The conventional approach, which we refer to as stepwise iterative refinement, is to find suitable values through a process of manual parameter tweaking by trial-and-error. Users supply parameter values, initialize and wait for algorithms to execute. The output is inspected, parameter values are changed, and the process repeated until satisfactory output is produced.

A second category of approach is to find optimal parameter values by using interactive visualization. Although some promising visualization methods for parameter optimization of image analysis algorithms have been proposed, they have not been widely adopted. This can be partly attributed to few examples of the application of such methods in real-world image analysis scenarios.

To address this, the primary contribution of this paper is a case study of a multidisciplinary team using interactive visualization to develop novel histopathology image segmentation software. We hypothesized that using visualization, as opposed to stepwise iterative refinement, our participants would: (1) analyze larger parts of parameter space; (2) gain a better understanding of the underlying algorithms; and (3) achieve higher quality results. Our study provides anecdotal evidence in support of our hypotheses and illustrates the potential usefulness of interactive visualization for parameter optimization purposes (albeit in a limited context). It also serves as an example of users' reactions to the introduction of interactive visual analysis into the established field of biomedical image analysis. As a secondary contribution, our work is an example of the role of case studies in iterative design. We discuss how an additional requirement was discovered and addressed. We also share preliminary results of how this changed user behavior.

## 2. Related Work

Many techniques exist for automating parameter optimization. Numerical optimization methods apply mathematical or statistical techniques to minimize (or maximize) an objective function defined over parameter space [Onw00]. Well-known examples include linear programming and gradient descent. Another trend is to apply artificial intelligence techniques (such as genetic algorithms) to optimization [AH97].

Automated techniques usually seek to minimize a single objective function. However, users often have multiple conflicting requirements. Moreover, end-users of image segmentation algorithms find it a daunting task to rigorously and mathematically formalize optimization criteria. This leads to human-in-the-loop approaches, which we describe below.

### 2.1. Stepwise Iterative Refinement

The most frequently encountered approach is to require users to search for optimal parameter values without additional support. Stepwise iterative refinement involves manual parameter tweaking by trial-and-error. It is a drawn-out procedure where users change parameter values and invoke a system or algorithm to produce the corresponding output. Output is judged qualitatively, input settings are changed, and the process repeated until satisfactory output is produced. For example, CellProfiler is a popular biomedical image analysis tool [CJL*06]. The incorporated algorithms are parameterized and users have to try different parameter values until they are satisfied with the quality of the output.

Stepwise iterative refinement is recognized as a bottleneck in parameter optimization [PBCR11,TWSM*11]: many iterations of the above process are typically required and users rely on memory recall to compare current output with previous results. Consequently, parameter space is inadequately explored, which negatively impacts the quality of output.

### 2.2. Interactive Visual Analysis

Several visualization methods have been developed to address the deficiencies of stepwise iterative refinement.

*Guided navigation.* Piringer et al. developed a technique for guided parameter optimization in a system called HyperMoVal [PBK10]. It shows multiple linked visualizations of the local neighborhood of a high-dimensional focal point in parameter space. This supports an iterative process where the user is guided to optimal parameter values by showing residual errors of estimated output of sampled parameters compared to known pre-computed output. In subsequent work, Berger et al. apply a similar approach for exploring parameter space [BPFG11]. The user is provided with visual guidance to potentially optimal regions in parameter space. Again, approximations of output are shown in visualizations of the local neighborhood of a user-selected focal point. Uncertainties associated with predicted output are also shown.

Torsney-Weir et al. present an approach to systematically explore the entire parameter space of a segmentation algorithm [TWSM*11]. Their system, Tuner, combines guided exploration with several linked visualizations of parameter space and output space. First, parameter space is sampled sparsely and segmentation outputs are computed for these points. Outputs are then evaluated with respect to a ground truth image (an optimal segmentation marked up by an expert) to compute quality measures. Next, a statistical model is used to estimate the quality of the entire parameter space. Regions likely to yield high quality outputs are highlighted. The user is guided to resample and investigate high-potential regions of parameter space until parameter values that yield suitable outputs are identified. In related work, Bergner et al. introduce an approach for interactive exploration of parameter space for multivariate simulation models [BSN*11]. The user is visually guided to discover regions of parameter space that are qualitatively different. There is also assistance for resampling regions of interest more finely.

The above methods are powerful but they require users to understand complex mathematical and statistical notions to interpret visualizations. Although there are niche users for which this is applicable (automotive engineers, for instance [TWSM*11]), end-users such as biomedical researchers are likely to find this challenging. The methods also rely on the existence of objective functions, quality measures, or ground truths, which are not always available.

*Interactive exploration.* A number of techniques enable users to explore parameter space and evaluate output qualitatively by visual inspection. Parameter space is typically high-dimensional, so standard multidimensional visualization techniques can be used [WB97]. Examples include parallel coordinates and scatterplot matrices [Ins85, Har75]. A number of techniques have been developed specifically for parameter visualization, however.

Several approaches, including those above, employ multiple coordinated visualizations. In the Influence Explorer, Tweedie et al. show histograms of input parameters and of outputs [TSDS95]. Users specify an active range for each histogram and data items for which one or more dimensions fall outside these ranges are dimmed. In later work, Tweedie and Spence extend the approach to prosection matrices [TS98]: 2D scatterplots, each with an associated third dimension for which users can specify an active range. Data items with one or more dimensions outside these ranges are hidden. With both methods, users discover relationships between parameters and outputs interactively. Dynamic querying enables them to explore different scenarios and identify parameter combinations that meet or fail requirements.

A few methods focus on changes in parameter values, and corresponding outputs, during the optimization process. Ma shows the parameter search process as a directed graph to help users find rendering parameter settings for computer graphics scenes [Ma99]. When parameter values are changed a new scene is rendered and a thumbnail representation is connected to the previous rendering with an edge. Callahan et al. use a similar approach for history management in VisTrails [CFS*06]: a visual history tree is maintained as users generate different visualizations of a dataset. Both methods use the visualization of the evolution of outputs as an external memory aid. Search paths are shown and existing outputs can be selected and refined further.

Some techniques emphasize the structure of parameter or output space. Jankun-Kelly and Ma use a spreadsheet metaphor to let users investigate the influence of rendering parameters on computer graphics scenes [JKM00]. Pairs of parameters define the x- and y-axes of a table and renderings that correspond to a pair of settings are shown at the intersection of the two coordinates. In Design Galleries, Marks et al. structure their visualizations on perceptually recognizable differences of outputs [MAB*97]. Parameter values are chosen so that outcomes are perceptually distributed. Several outputs illustrating these differences are shown to enable users to identify suitable regions of parameter space. Bruckner and Möller sample parameter space and apply a novel clustering method to identify key changes in the output space [BM10]. Both the clusters and output are visualized.

The visualization system considered in this paper supports interactive exploration of parameter space and was designed to analyze relationships between input parameters and outputs of biomedical image analysis algorithms. The approach does not rely on objective functions or ground truths as they cannot be assumed to exist in our target domain. An overview of this system is deferred to Section 3.2.

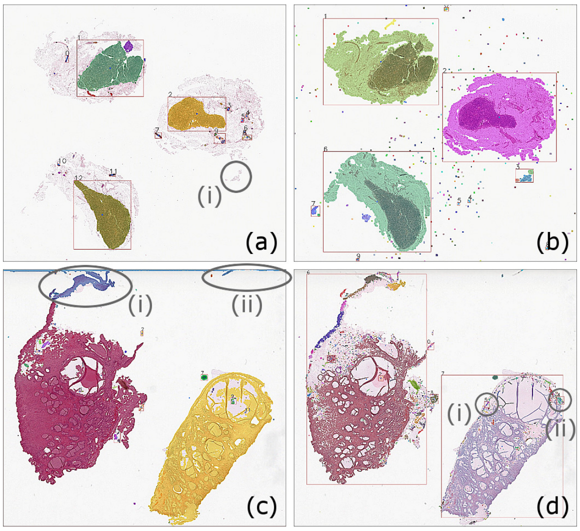

**Figure 1:** *Segmentation of histopathology images. Detected contiguous regions are filled with a random colour and outlined with a bounding box. (a) Poor quality segmentation of a lymph node section resulting from default parameter values identified through stepwise iterative refinement: multiple smaller parts of the three main contiguous regions are detected. (b) High quality segmentation of a lymph node section resulting from more optimal parameter values identified by interactive visual analysis with Paramorama. (c) Poor quality segmentation of a cervix section resulting from default parameter values: in addition to the coloured regions, the entire image is also detected as a contiguous region. (d) High quality segmentation of a cervix section resulting from more optimal parameter values identified by visual analysis. Histopathology images often contain noise resulting from sectioning ((a)(i) and (c)(i)), and artifacts on the glass slides ((c)(ii)). A subtle error in the original implementation of the algorithm, where smaller regions, such as (d)(i) and (ii) were not combined with larger regions that contain them, was also identified by visual analysis.*

## 3. Parameter Optimization for Image Analysis

To compare stepwise iterative refinement with interactive visual analysis, we conducted a case study. Before it is discussed, we first describe the software systems that we used.

### 3.1. Proprietary Segmentation Software

Histopathology images are very high resolution scans of microscopic-scale tissue slices (typically 80,000 × 100,000 pixels). Consequently, a user may be viewing morphological structures at a very high zoom level. This makes it difficult to navigate the images efficiently as context is lost. The software we considered is being developed as part of a set of methods to assist in (semi-) automatic navigation between distinct regions of tissue in histopathology images. It forms part of an ongoing initiative, independent of the present study, to develop virtual microscope software [TJOH*09].

The goal is for the software to accurately detect contiguous regions of tissue in an image. This is achieved by several image processing steps. For instance, a thresholding step distinguishes foreground and background based on pixel-level intensity differences. Each processing step requires the specification of values that indicate, for instance, the amount of thresholding to apply. As we describe in Section 4, a set of default values had been identified for these parameters by stepwise iterative refinement and were subsequently hardcoded in the software. For a particular input image, the output is an image where each detected region is filled with a color and marked by a bounding box (see Figure 1(a)).

### 3.2. Paramorama

We also considered a visualization prototype designed for parameter optimization of image analysis algorithms in a biomedical context. This technique differs dramatically from the software described in Section 3.1 in the way that parameter optimization is approached by providing interactive visual support. It is an example of using visual analysis for parameter optimization, as discussed in Section 2.2.

In previous work we presented a design study for interactive visual optimization of parameters for biomedical image analysis algorithms [PBCR11]. Our initial design was informed by interviews and discussions with domain experts at the Broad Institute of MIT and Harvard and observations of users at a hands-on introductory workshop on image analysis. Our technique was developed to visually analyze a Cartesian sampling of parameter space. This has the benefit of being conceptually simple: for each parameter a user wishes to sample, they simply specify a range and the number of samples to compute. To facilitate this, we originally implemented a sampling plugin for CellProfiler [CJL*06]. Our approach is not tied to CellProfiler or any particular algorithm, however. As we will demonstrate, users of other tools can extend these to sample any parameters of interest and analyze their results with our prototype. Requirements analysis revealed that users are usually interested in sampling 3–7 parameters at 3–7 intervals each [PBCR11].

As shown in Figure 2(a), we visualize sampled parameter space as a tree, using a node-link depiction oriented left-to-right. Every level represents a parameter and users can interactively change the order to suit their analysis needs. In the figure, the first parameter was sampled four times, shown by four second-level nodes (the root contains all samples). For each of these values, the second parameter was sampled four times, and so on. Users can identify, navigate to, and select particular combinations of parameter values. When they select regions of parameter space (subtrees in the node-link view) thumbnails of the actual image-based output are displayed in the detail view (Figure 2(b)). We developed a layout with the following properties [PBCR11]: each selection is shown in a distinct region; the top-to-bottom ordering of regions preserves the ordering of selections in the overview; the hierarchy of each selected subtree is shown by nesting; and within each region, the numerical order of sampled parameter values is shown left-to-right. The layout is parameterized and users can specify a parameter for which values are positioned top-to-bottom to make comparisons easier.

Users are often interested in contextual information in the form of a reference image to interpret the results of segmentation algorithms. Typically, they want to compare outputs to the original input image to determine whether correct objects, and not noise, were identified. We supply a reference image view as shown in Figure 2(c). When users move the cursor over any thumbnail, it is superimposed on the reference image. Our prototype provides standard interaction capabilities such as linked views and filters. A popular feature is to tag regions of parameter space that are associated with high- or low quality outputs. When a user tags regions, they are marked with a green (good) or magenta (bad) background in the overview and detail view (Figure 2(a) and (b)).

## 4. Histopathology Case Study

The software described in Section 3.1 requires parameters to be set to ensure accurate region detection. We hypothesized that the visualization method implemented in Paramorama (see Section 3.2) would enable users to: (1) consider larger parts of parameter space; (2) better understand their algorithms and the influence of parameter values; and (3) achieve higher quality segmentation. We observed three participants as they used Paramorama to analyze the results of the segmentation software using sampled parameter values.

### 4.1. Method

*Participants.* Adam is an undergraduate research assistant who had implemented the histopathology segmentation algorithm described. Bob is an academic member of staff and an active member of the image processing research community. He has developed several biomedical segmentation algorithms in the past and served as Adam's supervisor. Charles is a practicing histopathologist with deep domain knowledge in the analysis of histopathology images. He has been involved in several research projects where image analysis methods were applied in a histopathology context. Charles served as a domain expert during the development of the algorithm. None of the participants had prior experience with our visualization prototype. All were initially skeptical about the suitability of interactive visualization methods for the analysis and optimization of segmentation results.

*Materials.* Three tissue types were considered: biopsies from the oesophagus of patients with Barretts' esophagus, resections of the cervix, and excised lymph nodes. For each type, two images were selected. The first was a "textbook" example with one to three clean contiguous regions of tissue. The second image, which contained more noise, was emblematic of what often happens in practice. Histopathology images are high-resolution scans of thin sections of tissue fixed to glass slides. Noise is introduced when tissue tears, separates, and distorts during the sectioning process (see Figure 1(a)(i) and (c)(i)), or by artifacts on the glass (Figure 1(c)(ii)). All sections had been treated with hematoxylin and eosin stain (H&E), which is widely used in histopathology. It colors cell nuclei blue while cytoplasm is colored pink/purple. The study was conducted on a desktop workstation (2.4GHz CPU; 12GB RAM) running Windows 7 with a 30-inch flat panel monitor (2,560 × 1,600 pixels; 60Hz).

*Procedure.* An initial discussion with Adam and Bob revealed the overall goal of the segmentation software being

Parameters
Order in hierarchy:
Threshold
Smooth
Dilate
LowerBound
Expand

Filter
Filter based on:
● Parameters ○ Tags
Parameter filters:
Threshold 140 230
Smooth 1 13
Dilate 1 4
LowerBound 11 20
Expand 2 5
Tag filters:
☑ Positive ☑ Negative
☑ Untagged
Apply:
● Gray out ○ Hide

**Figure 2:** *Visual analysis of parameter space with Paramorama. (a) Sampled parameter space is visualized as a tree where, left-to-right, every level represents a parameter. (b) For selected regions in parameter space, the corresponding image-based output is shown as thumbnails in the detail view. Users can tag results as good (green) or bad (magenta). (c) A reference image, typically the original input image, is supplied to visually compare the results to. The visualized data is of segmentation results for a histopathology image of lymph node tissue.*

developed and the research context in which this was happening (see Section 3.1). It was a challenge to identify how the implementation of the algorithm had been parameterized. After discussions with Adam and Bob, it was decided that some of the hard-coded values in the algorithm might better be considered as parameters and that different values for these might be more appropriate under different circumstances. Adam's approach had been to run the algorithm on a single input image and to identify suitable parameter values by stepwise iterative refinement. Once identified, these values were hard-coded in the implementation of the algorithm. Adam and Bob were confident about the robustness of these values and were not convinced that they could be improved on by exploring the parameter space further.

After a code-review with Adam, five parameters were identified for closer scrutiny. These were *smoothing* (how much noise reduction to apply), *thresholding* (the pixel intensity at which to differentiate foreground and background), *dilation* (how much to expand the foreground to "fill" holes), *expansion* (how much to expand foreground regions before detecting intersections), and *merging* (a threshold for discarding or combining intersecting regions). Adam modified his code so that it could be called with arguments to specify the parameter values to use. He then wrote a script to perform Cartesian sampling of the parameter space and to call the algorithm with all unique combinations of the sampled parameter values. Next, suitable intervals and the number of samples to compute over each interval were identified: parameters were sampled four times each, yielding 1,024 unique combinations. Corresponding segmentation output for the six input images were computed overnight.

After a training session with our prototype Adam was given an opportunity to identify regions of parameter space that produce high-quality segmentation results. In a second session, Bob was given training to use our prototype and then asked to review Adam's results. Finally, in a third session, the functionality and features of our prototype were demonstrated to Charles before being given an opportunity to try it. All participants were comfortable using Paramorama after 30 minutes of hands-on training.

### 4.2. Results

*Adam.* Our first participant required 10-30 minutes to analyze each dataset for a total analysis time of 1 hour 21 minutes for all six. The first two datasets, respectively, required

25 and 32 minutes but this dropped to between 9 and 11 minutes for the final three datasets.

Using the node-link visualization of parameter space (see Figure 2(a)), subtrees were systematically selected starting from the top and working to the bottom. Thumbnail versions of the corresponding outputs were viewed in the detail view (Figure 2(b)). In most cases Adam tagged all displayed thumbnails based on an initial impression formed by scanning the thumbnails ("good" or "bad"). Tagging was done in one operation by using the nested representation of the detail view to select all outputs. Most regions of parameter space that Adam thought yielded poor results were identified like this. Individual results of which he was uncertain were viewed in more detail using the reference image view. This often resulted in flicking "bad" tags to "good" (and vice versa). We noticed that Adam devised a rule-of-thumb for each of the six input images during his analysis. This involved scanning the output to determine whether key regions had been identified. This was often vocalized, for example: "yes, the five major regions are correctly detected" or "too many regions are identified, it's picking up noise".

Adam made a number of important discoveries. For all six input images, more optimal parameter values that differed from the hard-coded values were identified. Using the number of segmentations as a crude quality metric, the default parameter values resulted in 21 regions being erroneously segmented for Barrett's noisy, 22 for cervix noisy, 26 for lymph clean, and 16 for lymph noisy. Visual inspection also revealed that the boundaries of regions detected for Barrett's clean and cervix clean were closer to what a human expert would have picked. In particular, two parameters (*smoothing* and *thresholding*) were discovered to have an important influence on the quality of the outputs produced by the segmentation algorithm. Figure 1 contrasts the results of applying the segmentation algorithm to two of the datasets with the original hard-coded parameter values (Figure 1(a) and (c)) versus the values identified with Paramorama (Figure 1(b) and (d)). Even casual visual inspection shows that there is a vast improvement in quality for the newly identified values. Adam also came to a more general conclusion: suitable parameter values are extremely context sensitive. Each of the six images required a different set of parameter values to yield the best results. This convinced Adam that different types of tissue and the amounts of noise in the images require different parameter values to ensure high-quality segmentation.

*Bob.* Upon reviewing Adam's tagged results, Bob did not perform an exhaustive analysis of parameter space, but focused on a few regions that had been mostly tagged as "good" or as "bad". Bob next scanned the thumbnails for these regions in the detail view. For many results he deemed the thumbnails sufficient to confirm his agreement with the categories that had been assigned by Adam. Although not as vocal about the number of regions that the algorithm had segmented, Bob did pay attention to object count. Many results were rejected by noting, for example, "far too many artifacts". However, there were a number of individual results where Bob proceeded to view the segmentation results in more detail using the reference image view (see Figure 2(c)).

By reviewing his colleague's work, Bob identified a subtle logic error. The algorithm detects small contiguous regions and should progressively combine these to form larger regions, provided certain criteria hold. However, in many cases overlapping regions of dissimilar size were not being combined (see Figure 1(d)(i) and (ii)). Bob concluded that either the heuristics for deciding when to combine smaller regions were insufficient, or the implementation was incorrect. This sparked follow-on discussions between Adam and Bob that confirmed the existence of a logic error. It also led to a code review where additional implementation errors were discovered. The logic and implementation errors were flagged as high priority changes for a next release of the software.

*Charles.* When Charles reviewed Adam's results, he spontaneously changed the order of parameters in the node-link representation of parameter space. He explained that his aim was to identify those parameters that have the most "striking" influence on segmentation quality, using Adam's tags as an indication of quality. He remarked that our prototype made these relationships "very clear". Using an approach similar to that of Bob, many cases were found where Charles disagreed with the initial tagging of the outputs. This was put down to inexperience and a lack of in-depth domain knowledge on the part of Adam. Charles recognized this as a common problem when trying to get results quickly and emphasized the need for more expert reviews in the future.

### 4.3. Discussion

The above results provide anecdotal evidence to support our hypotheses (within the context of our study). First, participants explored larger parts of parameter space using our visualization prototype than they had previously using stepwise iterative refinement. All participants agreed that it would have been prohibitively time-consuming to analyze the number of parameter combinations they considered with Paramorama using their conventional approach. Second, visualization helped all participants to achieve new insights and learn more about the relationships between input parameter values and the resulting output. They were also more confident in their understanding of the algorithm than before. Third, using interactive visualization, participants achieved higher quality output that differed from those produced by hard-coded parameter values.

In our experience, our case study is a good example of the reaction of users to the introduction of visual analysis as an alternative to an established approach (stepwise iterative refinement of parameters for image analysis algorithms, in this case). Despite initial skepticism, participants were soon able to apply the new method to great effect. As a case in point, an important logic error and implementation errors were discovered using our visualization prototype. These had previously gone undetected despite the analysis of segmentation output and the identification of what were believed to be optimal parameter values. Also, all participants were far less certain about the robustness of the algorithm and the hard-coded parameters after using Paramorama than they had been originally. This suggests that visual parameter optimization may be able to facilitate critical reflection better that stepwise iterative refinement.

Visual analysis also highlighted flaws in participants' assumptions and quality control during algorithm implementation. In particular, it emphasized the gap that exists between "good" results for different stakeholders (with technical versus domain knowledge). Our prototype facilitated discussion between stakeholders that would have been difficult without the ability to visually compare and refer to different outputs interactively. The threefold drop in analysis time by Adam over the six datasets suggests that there is an associated learning effect, but once users become familiar with visualization tools, such as Paramorama, they should be able to analyze large parts of parameter space in shorter periods.

The case study also revealed an important shortcoming of our approach. In terms of analysis strategy, Adam systematically considered all outputs corresponding to sampled parameter values. However, he often referred to the number of objects detected when judging whether an individual result is acceptable. The other two participants also mentioned object count while analyzing the outputs. From this we concluded that, had a facility been available to select or filter results based on object count, Adam would probably not have analyzed all outputs exhaustively but would have discarded large parts of parameter space where associated object counts were unreasonable. We also suspect that such a facility would have enabled Bob and Charles to target their detailed analysis to smaller subsets of the generated output. When we raised the issue, our participants agreed on the potential advantages of augmenting our approach with derived measures such as *object count* (the number of objects identified by segmentation) and *area occupied* (the sum of regions occupied by detected objects). They felt that this would allow users with a "gut feeling" of what suitable quality involves to identify high-potential parts of parameter space.

## 5. Follow-on Work

To investigate the role of derived measures, we extended our prototype to show a histogram of the distribution of values for each derived measure associated with the data (see Figure 3(a)). Inspired by the work of Tweedie et al. [TSDS95], this metric view is linked with all other views (Figure 3(b)–(d)): when the user selects one or more bars in a histogram the corresponding regions of parameter space are selected in the overview and the output shown in the detail view.

We were interested to see whether, and how, users who had not previously used our prototype would apply this new feature. To do so, we performed a preliminary study with eight students majoring in biological sciences. Because histopathology images were considered too specialist, we used the sampling plug-in we had developed for CellProfiler to compute segmentations for two 640 × 640 pixel photomicrographs of human osteosarcoma cells (cancerous bone tumor) [PBCR11, Bro]. Both images had been stained with DRAQ, which highlights cell nuclei as light gray on a dark background. Five parameters were sampled four times each to yield 1,024 different outputs. After training, each participant was presented with the results of one of the two images and given 10 minutes to find a region of parameter space where cell nuclei are most accurately detected.

All participants used the metric view to converge on a region of parameter space where they thought the object count was correct (see Figure 3(a)). In most cases, they started with a low guess for object count, selected the corresponding histogram in the metric view, and scanned the output thumbnails in the detail view (Figure 3(c)). Repeating this process, they revised the object count upward until settling for a value. As observed during the case study (Section 4.2), participants then considered the thumbnails in the detail view to analyze individual outputs (Figure 3(d)). Parameter values were inspected with the node-link view (Figure 2(b)).

This is a positive result that shows an improvement on our previous design, but there are further opportunities to explore. First, the inclusion of several derived measures could assist users in further reducing the subset of outputs to review, as with faceted search [Tun09]. Other measures, such as the total area covered by identified objects, could be used for this (our prototype already caters for an arbitrary number of measures). Second, the output of an objective function for each point in parameter space could be shown to more closely resemble guided navigation (see Section 2.2). We note, though, that all domain experts we interviewed were reluctant to introduce more complex measures. A third option is to visualize the distribution of outputs per parameter for a meaningful measure (using histograms, for example) and to enable users to directly filter the outputs to view in more detail using a scented widgets approach [WHA07].

## 6. Conclusion

The presented case study serves as an example of the potential usefulness of interactive visualization to identify suitable parameter values for biomedical image analysis algorithms. As hypothesized, using our visualization prototype, participants were able to: (1) analyze larger parts of parameter space; (2) obtain a better understanding of the underlying algorithms; and (3) achieve higher quality results than they had previously with stepwise iterative refinement. The study also serves as an example of users' reactions to the introduction of interactive visual analysis into the established field of

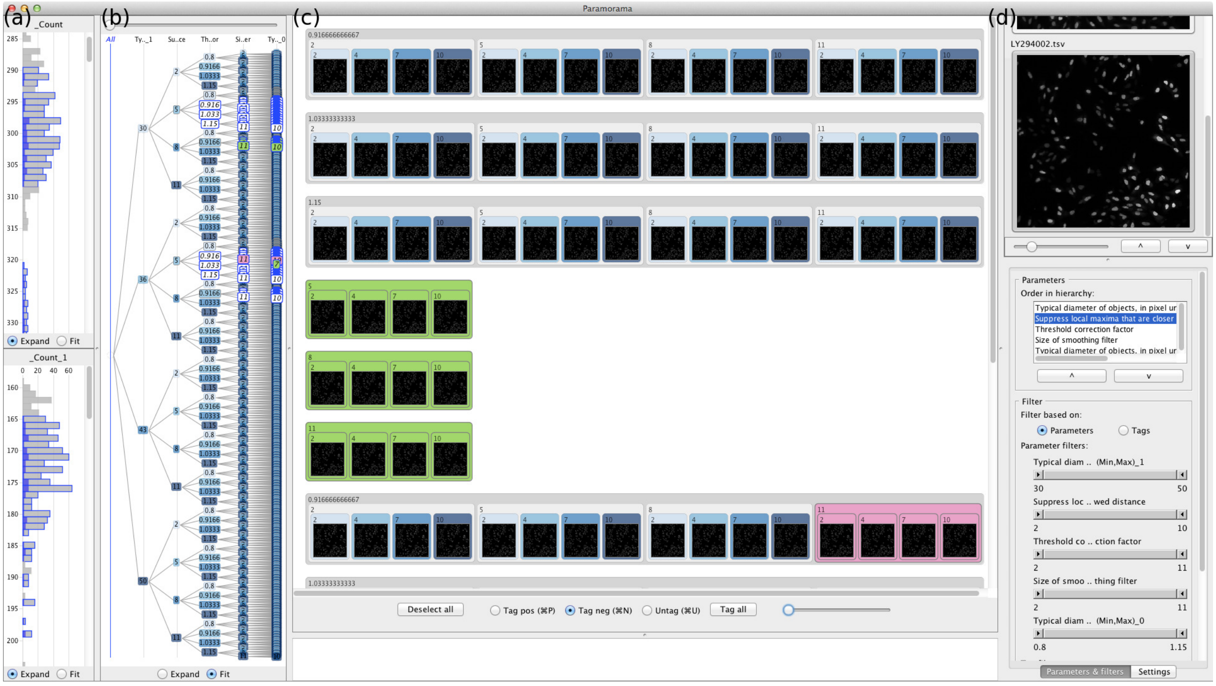

**Figure 3:** *(a) Interactive histograms that show the distribution of output for one of more derived measures were added to our prototype after a case study revealed this as a requirement. These are linked with (b) the parameter space view, (c) the detail view, and (d) the reference image view. The visualized data is of segmentation results for photomicrographs of osteosarcoma cells (cancerous bone tumour).*

biomedical image analysis. Despite initial skepticism, participants discovered an unknown logic error in their algorithm and errors in its implementation.

Our work also serves as an example of how case studies can serve as an important part of iterative design. Collaboration with experts revealed an important additional requirement: facilitating analysis of derived measures. Our preliminary results suggest that interactive visualization of derived measures further reduces user effort during parameter optimization. Based on this initial work, we anticipate that users will benefit further from methods that help them effectively select an even smaller subset of outputs for detailed analysis. Our prototype is available for download at [Pre].

We anticipate that the visual analysis design principles implemented in Paramorama should generalize to different image analysis algorithms and input images (in biomedicine and beyond). Indeed, the case study and follow-on work discussed in this paper were based on different segmentation algorithms and considered different input images (histopathology versus photomicrographs). To investigate this and to generalize the findings summarized above, more empirical evidence is needed. This presents a number of challenges.

Rigorous user evaluation studies require a large participant sample size while biomedical data sets encountered in the real world often require scarce specialist skills to interpret and analyze. Moreover, our findings suggest that participants would require sufficient time for training and we anticipate that the value of interactive visualization methods will emerge with repeated use over extended periods of time. This implies a longer evaluation timeline than feasible with a typical laboratory-based investigation. Designing and implementing user evaluation studies that meet all these requirements is a significant challenge similar to those encountered by other visualisation researchers [Pla04]. In this light, a more realistic goal may be for researchers to conduct more longitudinal investigations, to accumulate, and to present more anecdotal evidence, such as the present study [SP06].

## Acknowlegements

We thank Anne Carpenter and Mark Bray (Broad Institute of MIT and Harvard) for crucial input and John Lau (University of Leeds) for his collaboration. This work was funded through WELMEC, a Center for Excellence in Medical Engineering funded by the Wellcome Trust and EPSRC, under grant number WT088908/Z/09/Z.

## References

[AH97] Ansari N., Hou E.: *Computational Intelligence for Optimization*. Springer, 1997. 68

[BM10] Bruckner S., Möller T.: Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1468–1476. 69

[BPFG11] Berger W., Piringer H., Filzmoser P., Gröller E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum 30*, 3 (2011), 911–920. 68

[Bro] Broad Institute of MIT and Harvard: Broad Bioimage Benchmark Collection, SBS Bioimage CNT. http://www.broadinstitute.org/bbbc/sbs_bioimage_cnt.html. Last visited 31 March 2012. 73

[BSN*11] Bergner S., Sedlmair M., Nabi S., Saad A., Möller T.: Paraglide: interactive parameter space partitioning for computer simulations. *Computing Research Repository abs/1110.5181* (2011). 68

[CFS*06] Callahan S. P., Freire J., Santos E., Scheidegger C. E., Silva C. T., Vo H. T.: Managing the evolution of dataflows with VisTrails. In *Proceedings of the International Conference on Data Engineering Workshops* (2006), pp. 71–75. 69

[CJL*06] Carpenter A. E., Jones T. R., Lampbrecht M. R., Clarke C., Kang I. H., Friman O., Guertin D. A., Chang J. H., Lindquist R. A., Moffat J., Golland P., Sabatini D. M.: CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology 7*, R100 (2006). 68, 70

[Har75] Hartigan J. A.: Printer graphics for clustering. *Journal of Statistical Computation and Simulation 4*, 3 (1975), 187–213. 68

[Ins85] Inselberg A.: The plane with parallel coordinates. *The Visual Computer 1* (1985), 69–91. 68

[JKM00] Jankun-Kelly T. J., Ma K.-L.: A spreadsheet interface for visualization exploration. In *Proceedings of the IEEE Conference on Visualization* (2000), pp. 69–76. 69

[Ma99] Ma K.-L.: Image graphs - a novel approach to visual data exploration. In *Proceedings of the IEEE Conference on Visualization* (1999), pp. 81–88. 69

[MAB*97] Marks J., Andalman B., Beardsley P. A., Freeman W., Gibson S., Hodgins J., Kang T., Mirtich B., Pfister H., Ruml W., Ryall K., Seims J., Shieber S.: Design Galleries: a general approach to setting parameters for computer graphics and animation. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques* (1997), pp. 389–400. 69

[Onw00] Onwubiko C.: *Introduction to Engineering Design Optimization*. Prentice-Hall, 2000. 68

[PBCR11] Pretorius A. J., Bray M. A. P., Carpenter A. E., Ruddle R. A.: Visualization of parameter space for image analysis. *IEEE Transactions on Computer Graphics and Applications 17*, 2 (2011), 2402–4211. 68, 70, 73

[PBK10] Piringer H., Berger W., Krasser J.: HyperMoVal: interactive visual validation of regression models for real-time simulation. *Computer Graphics Forum 29*, 3 (2010), 989–992. 68

[Pla04] Plaisant C.: The challenge of information visualizations evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (2004), pp. 109–116. 74

[Pre] Pretorius A. J.: Paramorama website. http://www.comp.leeds.ac.uk/scsajp/applications/paramorama/. Last visited 1 September 2012. 74

[SP06] Shneiderman B., Plaisant C.: Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the AVI Workshop on Beyond Time and Errors* (2006), pp. 1–7. 74

[TJOH*09] Treanor D., Jordan-Owers N., Hodrien J., Quirke P., Ruddle R. A.: Virtual reality powerwall versus conventional microscope for viewing pathology slides: an experimental comparison. *Histopathology 55*, 3 (2009), 294–300. 69

[TS98] Tweedie L., Spence R.: The prosection matrix: a tool to support the interactive exploration of statistical models and data. *Computational Statistics 13*, 1 (1998), 65–76. 68

[TSDS95] Tweedie L., Spence B., Dawkes H., Su H.: The influence explorer. In *Proceedings of the Conference on Human Factors in Computing Systems* (1995), pp. 129–130. 68, 73

[Tun09] Tunkelang D.: Faceted search. *Synthesis Lectures on Information Concepts, Retrieval, and Services 1*, 1 (2009), 1–80. 73

[TWSM*11] Torsney-Weir T., Saad A., Möller T., Weber B., Hege H. C., Ver-bavatz J. M., Bergner S.: Tuner: principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Computer Graphics and Applications 17*, 2 (2011), 1892–1901. 68

[WB97] Wong P. C., Bergeron R. D.: 30 years of multi-dimensional multivariate visualization. *Scientific Visualization: Overview, Methodologies, and Techniques* (1997), 3–33. 68

[WHA07] Willett W., Heer J., Agrawala M.: Scented widgets: improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1129–1136. 73