Research Article

TissueWand, a Rapid Histopathology Annotation Tool

Martin Lindvall[1,2,3], Alexander Sanner[1], Fredrik Petré[1], Karin Lindman[4,7], Darren Treanor[2,5,6], Claes Lundström[1,2,3], Jonas Löwgren[3]

[1]Sectra AB, Research Department, Linköping, Sweden, [2]Center for Medical Image Science and Visualization, Linköping University, Linköping, Sweden, [3]Department of Science and Technology (ITN), Linköping University, Norrköping, Sweden, [4]Department of Clinical Pathology, Region Östergötland, Linköping, Sweden, [5]Department of Cellular Pathology, Leeds Teaching Hospitals NHS Trust, Leeds, UK, [6]University of Leeds, Leeds, UK, [7]Department of Biomedical and Clinical Sciences, Linköping University, Linköping, Sweden

Submitted: 10-Jan-2020 Revised: 23-Mar-2020 Accepted: 19-May-2020 Published: 21-Aug-2020

Abstract

Background: Recent advancements in machine learning (ML) bring great possibilities for the development of tools to assist with diagnostic tasks within histopathology. However, these approaches typically require a large amount of ground truth training data in the form of image annotations made by human experts. As such annotation work is a very time-consuming task, there is a great need for tools that can assist in this process, saving time while not sacrificing annotation quality. **Methods:** In an iterative design process, we developed TissueWand – an interactive tool designed for efficient annotation of gigapixel-sized histopathological images, not being constrained to a predefined annotation task. **Results:** Several findings regarding appropriate interaction concepts were made, where a key design component was semi-automation based on rapid interaction feedback in a local region. In a user study, the resulting tool was shown to cause substantial speed-up compared to manual work while maintaining quality. **Conclusions:** The TissueWand tool shows promise to replace manual methods for early stages of dataset curation where no task-specific ML model yet exists to aid the effort.

Keywords: Annotation, digital pathology, usability, user interface design

Introduction

In recent years, machine learning (ML) and especially deep learning with convolutional neural networks have achieved near-human–level performance for certain medical imaging diagnostic tasks in experimental settings.[1,2] For a majority of these tasks, the training of ML models requires human expert annotators to perform extensive labeling and data curation.[3] This annotation effort is one of the major bottlenecks in realizing the potential of new ML techniques[3] for real-world applications. Typically, manual annotation efforts can take months of full-time work to complete.[4]

While automatic segmentation of arbitrary images remains an unsolved problem, a middle ground offered by the use of interactive assistive tools is an active research topic.

In digital histopathology, annotation to facilitate ML typically involves reviewing gigapixel-sized images of tissues and assigning multiclass labels to delimited regions of the image. Annotation in this setting presents novel challenges for the design of interactive semi-automatic tools. Whole-slide images (WSIs) might be up to 1000 megapixel (MP), but a typical computer monitor can only show 2–4 MP. Features to be annotated can be large (mm) – such as a tumor or region of tissue – or tiny (10 um) – such as a single cell. Borders between classes are not solely defined by a clear change in contrast but also defined by changes to the overall texture. Sometimes, no clear borders exist.

We have developed a tool for efficient interactive annotation in digital histopathology, named TissueWand. Apart from presenting the tool itself, this paper also describes the explorative design approach to discover requirements and iteratively construct the tool, since we believe that the insights gleaned in this process can be informative for future similar efforts. Moreover, we present the results from a small user

Address for correspondence: Mr. Martin Lindvall, Sectra AB, 58330 Linköping Sweden.
Email: martin@ixd.ai

Access this article online

Quick Response Code:

Website: www.jpathinformatics.org

DOI: 10.4103/jpi.jpi_5_20

How to cite this article: Lindvall M, Sanner A, Petré F, Lindman K, Treanor D, Lundström C, *et al*. TissueWand, a rapid histopathology annotation tool. J Pathol Inform 2020;11:27.

Available FREE in open access from: http://www.jpathinformatics.org/text.asp?2020/11/1/27/292720

study evaluating efficiency and quality by comparing our final version to manual annotation.

This paper first reviews previous work where user interactions have been used to facilitate annotation, labeling, or segmentation. Next, we describe and motivate our iterative design-based research approach – to jointly explore algorithmic and interaction design through high-fidelity prototypes in-use. We then give an iterative account of our insights during the prototyping phases. Next, we present another part of our results, i.e., the findings from our quantitative user study. Finally, we discuss the relevance of our work to ongoing and future research.

Related work

In this section, we summarize prior work with a focus on user interactions that can be used to produce annotations or provide other methods to aid in gathering training data for ML. We describe work in interactive segmentation, ML, and studies that these techniques were specifically applied to histopathology.

Interactive segmentation

Interactive segmentation stems from work in automatic segmentation. In automatic segmentation, the goal is to assign each pixel in a given image to its constitutive category without supervision. Fully automatic segmentation is difficult partly because the problem is underspecified, i.e., there exists many possible categorization systems to use when assigning pixels to categories. In interactive segmentation, computational methods are combined with human operators.

User interactions for interactive segmentations have typically been based on common interaction motifs, which can be described broadly as boundary or region-based methods.[5] In boundary methods, users' input corresponds to delineating the boundaries between categories, such as click-dragging along edges.[6,7] In region-based methods, users typically click-drag to paint categories to signify pixels that lie within the segments.[8,9] Typically, as a result of initial inputs, the resulting full-image segmentation is visualized as a shaded overlay or as polygons delimiting categories.

If the user interface supports iterative refinement, the user can update the resulting segmentation, receive a new proposal, and continue this process until a desirable result is reached. Iterative refinement requires strong usability considerations to avoid user frustration: using a system that requires more iterations but allows easier iterative improvements has been shown to be preferred over a system with a better initial guess, but where improvements are difficult to achieve.[5]

Notable variations to boundary, region-based, and iterative refinement exist. For instance, in the refinement stage the input method might change, e.g., first drawing a box and then allow updates through painting categories.[10] The resulting segmentation from region-based input can be limited by distance from the initial seeds, allowing the user more control at the expense of number of operations performed.[11] To speed up processing at interaction time, a precomputed oversegmentation can be used with algorithms applied to these larger segments rather than on raw pixels.[9]

Interactive machine learning

Interactive ML (IML)[12] was originally formulated as a class of systems where users iteratively train a machine learner while receiving feedback on model performance after each interactive step. The Crayons System[12] is visually similar to region-based interactive segmentation methods. The main difference from interactive segmentation is that the user's goal is reached by training a pixel classifier using ML methods. More recent formulations of IML have broadened the scope of such systems to encompass interactions for the user to control other aspects of the system, such as feature selection and model comparison.[13] Amershi *et al*.[14] suggested that IML systems should be rapid, focused, and incremental. The CHISSL System[15] was designed from these principles and found that rapid, real-time feedback and letting users control what to label next allowed users to select more beneficial instances for the model learning process compared to a sequential active learning condition.

Active learning and weak supervision

Some approaches adapt or modify supervised learning algorithms to allow the use of fewer, noisy, or less specific labels. In *active learning*, the requirement on the amount of training data is reduced by allowing the learning algorithm to choose which data, among unlabeled samples, should be annotated next.[16] *Weakly supervised learning* is a loosely defined family of methods where "weak" or "noisy" labels are used to only partly label instances, such as using image-level labels in multiple instance learning,[17] using large bounding boxes to highlight discriminatory regions,[18] or learning directly from scribbles.[19]

Several works present tools specifically aimed at speeding up object instance segmentation in parts or on a full image. One example is using ML to refine a user-provided bounding box to a polygonal annotation.[20,21] In the FluidAnnotation interface,[22] users can relabel or activate segments from a set of proposal segments from a pretrained model.

Semi-automatic annotation for histopathology

Tizhoosh and Pantanowitz,[23] reflecting upon the potential of digital pathology, have identified the need for labeled data as one of the biggest opportunities for digital pathology. Some tools facilitate making digital pathology more available by providing workbenches and collaborative spaces within which annotations can be made in a structured fashion, such as Cytomine,[24] QuPath,[25] SlideRunner,[26] and OpenHI.[27] There are also a few examples of annotation or delineation tools that move beyond manual drawing. The Ilastik tool provides ML-based segmentation that has been used for histological annotation.[28] Iterative correction approaches have been proposed using a tile-based interaction where users correct the output of a classifier by toggling coarse-grained tiles[29] and using additional sampling of uncertain regions.[30] Weak supervision has been used to let users draw rough boxes around the glands on hematoxylin and eosin (H&E)-stained images.[18]

Iterative refinement has been used together with ML to go from rough polygon annotations to more detailed ones.[31] As an alternative, some avoid manual annotation completely by restaining the slide such that the sought regions stand out and mapping the areas back to the original WSI.[32,33]

Methods

The aim of our work was to produce knowledge on how to design a novel interactive system that could contribute to more efficient histopathological annotation. To this end, we adopted a design-led approach where ideas on technical properties and interactive behaviors were sketched, refined, and assessed in an iterative process. In this section, we first present the context of the study including collaborators, design goals, and materials. We then give a characterization of the design process and intermediate iterations leading up to a final prototype and finally describe our evaluation method.

Study context

The current study ran in parallel with the efforts to annotate a large number of histopathological slides,[4] with the end goal of developing augmented intelligence methods for clinical practice. Professional relevance and flexibility to handle a wide range of tasks have been a priority. The related annotation project acted in early stages as the source for relevant use cases to our qualitative user experiments and in later iterations as a practical environment for evaluating prototype refinements. Evaluation sessions and co-design sessions took place on-site at the university hospital where the pathologists and physicians performing annotations resided.

Design goals

We set out to design a tool that could aid us in the costly ground-truth annotation process for subsequent ML studies. We had two primary design goals. First, the resulting annotations should be a suitable substitute for manual polygon-based annotations being done in the context of producing training data for a machine classifier, i.e., the human using the tool must be able to produce annotations at an accuracy comparable to fully manual annotations and the tool must be usable for a novel problem without assumptions of preexisting training data. Second, the total effort of using the tool must be lower than doing manual polygon-based annotations and the tool must be easily accessible to pathologist annotators.

Many previously proposed annotation methods tend to accept slightly lower quality as long as efficiency is boosted; the aim of our work is to maintain manual-level quality. Another aspect is that whereas some methods are designed for a specific domain problem, our goal is to develop a tool with generic applicability in histopathology.

Dataset and sources

Our goal was for the tool to be general to histological specimens. Its design was informed by annotator experiences from about 1000 clinically collected WSIs across several tissue types. The previously mentioned project for skin and colon annotation was a primary source of experience, and the tool design was also informed by annotation of breast tumors, lymph node metastases, and classification of ovarian carcinomas. The tool has primarily been tested on H&E-stained slides, which make up 95% or more of slides in clinical practice. Qualitative feedback during the iterative design process came primarily from three pathologists and one physician trained in histology.

Design method and rationale

In this study, we followed a constructive design research methodology with four major phases, each consisting of explorations, prototyping, and validations. The general trajectory of the process moved toward increasing professional and practical relevance. Design activities primarily consisted of user observations, high-fidelity prototyping, and semi-structured interviews.

In a typical design process, the designers seek to elicit understanding about possible software's fit by qualitatively evaluating prototypes of increasing fidelity. The general idea is to avoid overspending on technical development before the underlying solution space is well understood. In this study, the behavior of image analysis and semi-automatic segmentation support plays a key part in the overall interaction. Hence, since the human reaction to the specifics of the underlying algorithms plays a key role, we chose to do high-fidelity prototyping to a greater extent than what is usually found in design research studies.

While the design process will be described in-depth in the results section, a brief overview is provided here.

Phase 1: Finding balance between manual control and automatic support

We qualitatively evaluated existing approaches from interactive segmentation with pathologists and found region-based inputs with distance-constrained segmentations as the most promising approach.

Phase 2: Faster feedback and annotation support at different scales

Based on experiences that users had with our initial prototype, we improved our computational methods to support increased speed of feedback, varied granularity of annotations, and increased generalizability to multiple tasks and tissue types.

Phase 3: Rapid real-time refinement through local limitation

After Phase 2, we still struggled with users not experiencing enough control and getting stuck in correction cycles. To decrease the level of automatic support in favor of manual control, we settled on near-real–time (40 ms) segmentation predictions from the scribbles.

Phase 4: Tightly-coupled natural interaction through pen interaction

While the idea of working fast in rapid fine-grained steps seemed promising, we identified the control by computer mouse as potentially limiting and revised our design for a pressure-sensitive pen as the primary input method.

Evaluation method

We concluded our experiments with a quantitative comparison between manual polygon annotations and the prototype available at the end of Phase 3. Specifically, we chose to focus our assessment on the act of drawing the polygon annotations. Since we aim to provide a general tool applicable to new datasets, the goal is to accurately capture the user's intent. We assume that the practical challenges of eye–hand coordination involved in drawing are the most significant component by far of total task time when annotating tissue specimens.

Our approach was to compare the manual annotation of five images by a nonpathologist physician with much experience of manual annotation (Condition 1) with a test condition where two experienced pathologists independently annotated the same slides assisted by TissueWand (Condition 2), aiming to reach the same level of precision as the manual annotations.

The five images used in our assessment were selected randomly among low-complexity cases in a larger collection of manually annotated images. The pathologists in Condition 2 were trained for one hour before starting to work on the annotation task. For each of the five cases, the pathologist reviewed the labels and delineations of the manual annotations and then moved on to annotating a fresh copy of the image using TissueWand. The reason for this preparatory step was to focus our evaluation on the act of drawing rather than differences in annotation intent. After the assisted annotation session, the pathologists were interviewed on their experience of using TissueWand and how it compares to manual annotation practice.

Speed and quality were measured to compare assisted and manual annotations. Speed was defined as the time from the first drawing-related input event to the final input event. A type of annotation error that often is in focus is the misjudgment of the extent of a region compared to a medical ground truth. In our case, however, the relevant error is how well the user managed to follow the intended boundary. When scrutinized in large magnification, the intended boundary at any point is straightforward to find given the annotation made. We therefore defined an annotation quality measure as follows: 30 points was randomly selected from the delineation borders drawn on each image, and for each such point, the closest distance to the intended boundary was manually measured, and the average over the points was derived.

Results

Findings from design process

Our final prototype is a superpixel-based interactive tool that produces segmentations spreading from the user's cursor with a distance relative to applied stylus pressure with very low response times (~40 ms), creating a feeling of a tightly coupled human–machine system. This section describes the phases of the design process in terms of evaluations, insights, and decisions that led up to the final design. Detailed descriptions of the final features are described as part of the design process.

A summary of the functionality in the final prototype is provided in Figure 1.

Preparation: Observing the manual annotation process

We wanted to learn what fixed requirements exist for a semi-automatic tool with generic applicability in histopathology. By observing the annotation process in the related project,[4] we characterized the manual, polygon-based process as follows:

1. Extract and anonymize relevant WSIs
2. Review a small number of test slides and construct an annotation guide, a method for reproducible annotations for the current domain problem. The annotation guide includes applicable medical terms, inclusions/exclusion criteria, and target "resolution" of annotations
3. For each WSI
    a. Draw outlines over areas, disregard difficult boundary decisions for the time being
    b. Assign categories to areas
    c. Refine annotations in challenging areas.

There was a large variation in the size of the structures that were annotated. In the annotation of skin resections exemplified in Figure 2, both large and small structures were annotated (i.e., 10–1000 μm). The annotated colon mostly included large-level structures (1–10 mm) as shown in Figure 3. Other annotation tasks require the user to capture phenomena only occurring in small, dispersed regions apparent in high magnification, as exemplified in Figure 4 showing annotated tumor metastases in the colorectal lymph nodes.

Phase 1 – Finding balance between manual control and automatic support

To probe the space of possible solutions, we wanted first-hand knowledge of the viability of current interactive segmentation approaches when applied to histopathology. Specifically, we wanted to find a suitable position in the tradeoff between manual control and automated support.

To assess the available options, we applied a few publicly available tools to images from our dataset and inspected them in terms of user input mechanisms, levels of automatic support, and resulting visualization methods,[24,25,34] with their major mode of interaction and visualization depicted in Figure 5.

In the early exploratory work, we then chose to investigate three input interactions: single click, click and drag, and area selection. Output visualization modes included global and local results, e.g., whether an action results in the segmentation of the entire image as in Figure 5c or to a region bounded by some spatial relation to the original input[35] as in Figure 5a and b.

In a semi-automatic approach where the human repeatedly acts in a feedback loop with the machine-produced predictions, a low-fidelity prototype such as paper sketch would not provide the understanding we were after. Our initial prototype was thus a high-fidelity interactive prototype albeit limited to being usable under very controlled circumstances, namely a few preselected histopathology images with predefined categories.

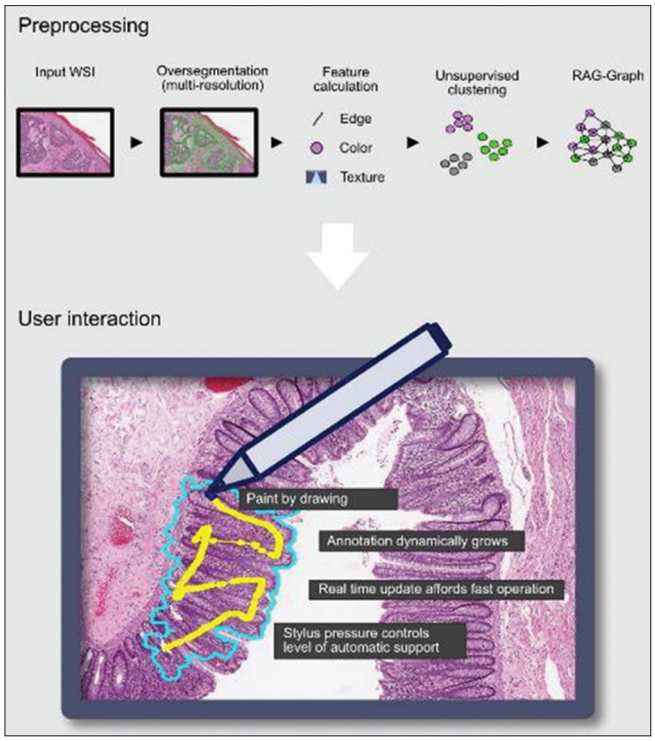

Figure 1: Overview of TissueWand functionality. Top: A preprocessing step prepares the image for efficient annotation by oversegmenting the image and precalculating features. The data are stored in a region–adjacency graph for fast access. Bottom: During user interaction, the user can annotate with a stylus. A local area around the point where stylus pressure is applied will also be annotated through an intelligent flood-fill controlled through the amount of pressure. In the illustrated use case, the central darker area of tissue (mucosa) must be segmented from the lighter pink tissue around it

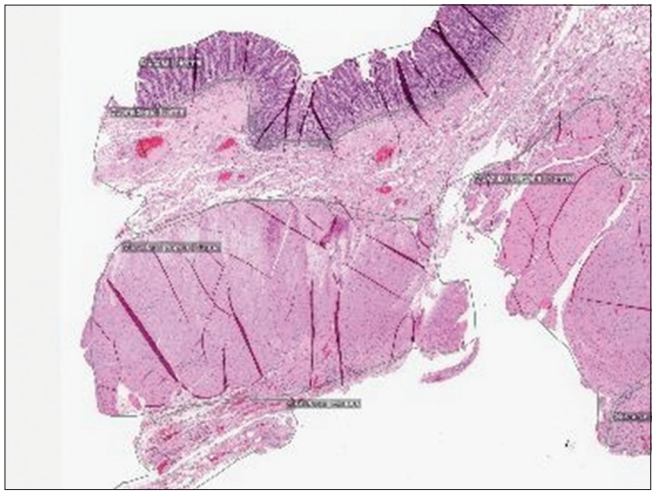

Figure 3: Example of annotation use case – Colon. Colon specimens annotated with a manual polygon-based tool as described in the previous study.[4] In this use case, only larger tissue structures readily visible at low magnification were annotated

From observing user behavior with the different input methods, we found a preference for click and drag. Single clicking had notable drawbacks; covering the large regions of a histopathology

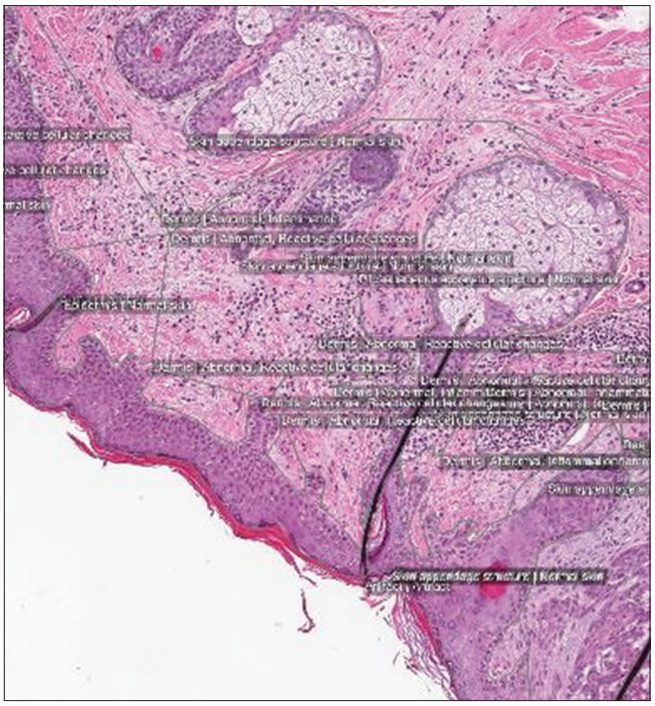

Figure 2: Example of annotation use case – Skin. Skin resection annotated with a manual polygon-based tool with nomenclature as described in the previous study.[4] It includes various parts of skin and features such as pathological changes and artifacts. The annotation alternates between overall tissue type and smaller structures

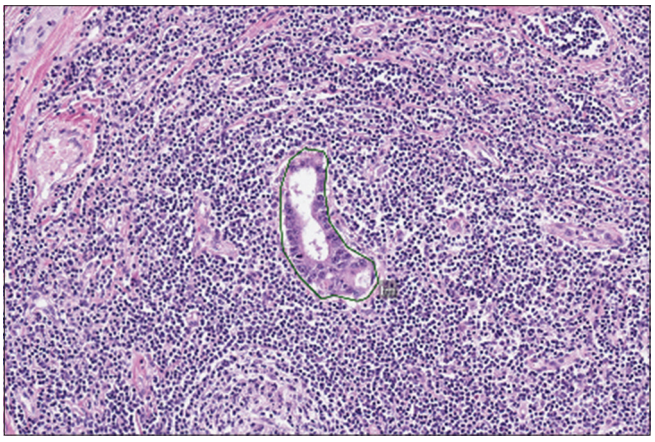

Figure 4: Example of annotation use case – Lymph node. Lymph nodes from colorectal cases. The annotated structures were small structures (a few tumor cells) in a much larger sample of unannotated (normal) lymph node tissue

images with only clicks means a very large amount of clicking if the level of automated support is low. While users did not express a dislike of area selection, we observed that by providing many options, users sometimes spent unnecessarily long times on the choice. Since we could not find a clear use case where the area tool was considerably more efficient than click and drag, we removed it from our list of candidate input methods.

Using global segmentation results with any of the input methods was rejected after observing user interaction with

early prototypes. While potentially powerful, returning a full segmentation after every input forces the user to revalidate the segmentation every time. For histopathological images, which are typically very large, this means that the time to search the entire image for errors to correct will overwhelm the user if there are many errors and the goal is to produce a segmentation of high quality. In other words, we observed that the cost of corrections was higher than the gains from increased automation. Since our annotation tool is meant to be used in a situation where the goal is to collect data to build a classifier that does not exist, a poor prediction and thus many errors are expected.

For our click-and-drag interaction, we aimed for fast response times. In general, choosing a suitable response time is challenging since it involves a tradeoff between speed and quality. An algorithm requiring more computation time usually gives a more accurate result. However, longer response times, and particularly variable response times, have an adverse effect on the quality of the interaction for the human operator. Put differently, better performance of an algorithm in a technical sense does not always imply better task performance of the joint human–machine ensemble.

One way to overcome this limitation is to divide the algorithm into two parts: (1) a slow part that preprocesses the image and (2) a fast part that is able to respond quickly. In our preprocessing stage, we presegment the image and then calculate features for these larger areas. For presegmentation, we used the simple linear iterative clustering (SLIC) algorithm to generate superpixels.[36] We chose the SLIC method since it is fast and generates superpixels of uniform size. For each superpixel, we then calculated a feature vector by applying binning to Hue, Saturation Value (HSV) colors and a texture feature based on local binary patterns (LBPs).[37]

For the online part, upon user input in the form of a line, we determined the hit superpixels and then returned a new set of superpixels as the result of the segmentation. In one rejected alternative, we simply returned the list of hit superpixels. While this gave the user a large amount of control, they also felt that they were not receiving enough automated support. In the version at the end of the phase, we compared the feature vectors of all hit superpixels, plus their immediate neighbors, to the superpixel at the start of the line using simple Euclidean distance. If above a preset threshold, the superpixel was included in the result.

The resulting interaction was not better than manual efforts in terms of time when evaluated with a few preselected testing images. When applied to novel images, the method transferred poorly due to being very sensitive to the size of phenomena to be annotated.

The main identified improvements at the end of the phase were:

- The tool would need to support annotation at different granularities. The prototype, for reasons of both performance and user efficiency, relied heavily on rather large superpixels and thus failed unrecoverable when those few superpixels did not correctly capture the user's annotation intent
- We would need to address the user getting stuck in "correction loops," where the user repeatedly tries to undo or erase a failed result only to get the same error again or some other part of the image wrong
- Users were annoyed over the system not including regions that were specifically drawn over. This was a result of considering only the feature of the superpixel at the beginning the user's line.

Phase 2 – Faster feedback and annotation support at different scales

The efforts in this phase mainly focused on making annotation using the click-and-drag interaction from Phase 1 possible at dynamic levels of detail while also having a relatively fast system response time (~100–500 ms) when used in a web-based client-server architecture. We also sought to address issues with the feature comparison not correctly capturing the user's intent when they had drawn over areas they wanted to include.

To support annotating at both high and low resolution, while having fast response times during interactive annotation, we added the precomputation of a pyramidal superpixel structure using the SLIC algorithm, using relatively small region size (10 pixels) for each layer. The small size ensures that phenomena are over-segmented. This superpixel pyramid represents regions at increasing levels of detail, as depicted in Figure 6.

The level in the superpixel pyramid that is chosen on user input is only indirectly controllable by the user through the zoom level at which the image is currently viewed. We found this simpler to grasp than having a separate "size" setting that could ambiguously be interpreted as the size of the brush. After some tuning, we settled on trying to select the level to have a roughly constant size superpixel relative to the size of phenomena viewed. The selected level is the closest one which will result in superpixels of 40 pixels wide given the current magnification, but never larger.

To address the issue with not capturing intent due to how features were compared, we investigated the changes to which superpixels to consider, how to compare them, and the effectiveness of underlying features. We evaluated features described in two related works[29,38] and found that combining HSV and LBP features with ray features[39] gave a good balance of predictability and accuracy when applied to our prototype. For added efficiency, we also precomputed a region adjacency graph that makes it efficient to consider a variable amount of neighbors to hit superpixels in linear time.

In the final prototype of the phase, all superpixels intersecting with the user's line are included in the result. Additional superpixels are included by an affinity search procedure, combining similarity of features and spatial distance. The spatial distance used was the shortest number of superpixels

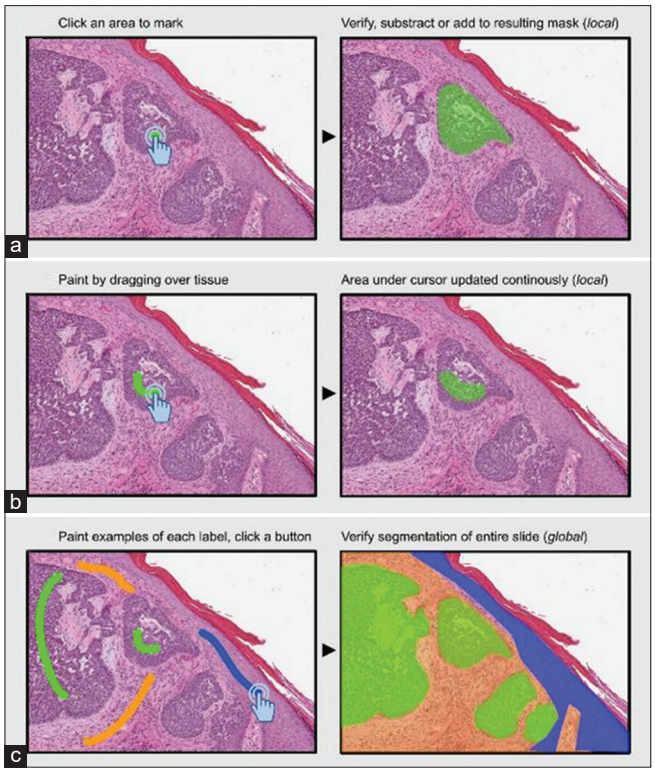

Figure 5: Operation of three other semi-automatic tools. (a) The wand tool in CytoMine[24] is a single-click flood-fill from the point of the mouse cursor. The result is local, limited to an area near the mouse. (b) The wand tool in QuPath[25] is a click-and-drag flood-fill using a smoothing preprocessing. Updates are received in real time as the user drags. Results are local. (c) In Ilastik,[34] the user provides examples of each label by click-dragging. After a button click, the user sees the result of labeling the entire image, a global result

from any input superpixel, and the similarity was computed as the cosine similarity between a candidate superpixel feature vector and the median feature vector of the input. The search is essentially a flood fill, growing from the directly hit superpixels. For details, see the pseudocode in Appendixes A.

After determining included superpixels, the resulting polygon is compared to the previous polygon for the given category, resulting in a number of changed edges and vertices. The list of polygon mutations is communicated back to the user's client in a compact binary format. The resulting interaction is depicted in Figure 7.

When the resulting prototype was evaluated, users could now capture both large and small structures, but the interaction was still experienced as a *trading of control between human and machine*, where the human waits for the machine response after drawing an area. After a short delay, the results are received and the user can make a correction, wait again, and then repeat the process. Typically, the user would be both intrigued and annoyed by the automatic assignment of the areas that were not specifically drawn over, sometimes resulting in long back-and-forth correction cycles without noticeable progress.

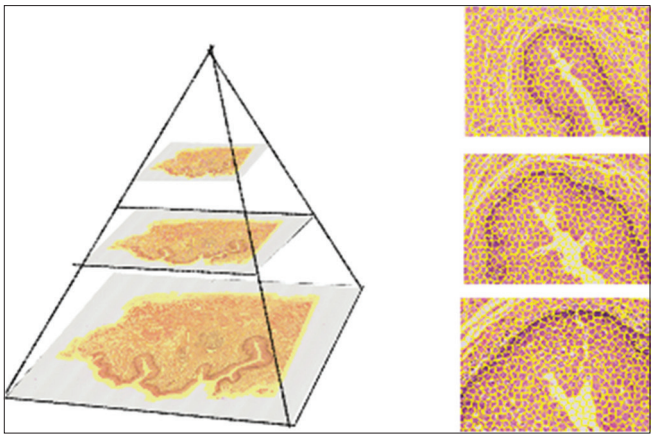

Figure 6: Pyramidal simple linear iterative clustering. A pyramidal structure of over-segmented areas at increasing resolution is computed in a preprocessing step using the simple linear iterative clustering algorithm iteratively. This ensures that system response times are kept low upon user input

We hypothesized that increasing the predictability of results could help avoid these correction cycles.

Phase 3 – Rapid real-time refinement

To offset the correction-cycle behavior apparent in the previous phase, we imagined a few options. We could increase the threshold with which superpixels are deemed similar and thus reducing the area that is automatically added to the user's stroke. This would increase manual control at the cost of automation, i.e., the total number of strokes required, which might be acceptable. Another option would be to make the threshold a parameter for the user to set, but that would introduce the user to the problem of finding a good setting before performing a potentially long scribble. We liked the idea of a controllable threshold since the threshold reflects the capability of the underlying features, whose effectiveness can vary between phenomena.

Among others, we imagined an interaction where the user selects an area in a two-step process, first by scribbling like before, but then receiving a slider that could be dragged to preview (in near-real time) segmentation results, as depicted in Figure 8.

Another option possible under the assumption that we could produce near-real–time predictions would be to give the user a preview instantly when the button is pressed, and updating this preview as the mouse button is dragged, using the mouse scroll wheel to change the threshold parameter at each point. When implemented as a prototype, we realized that the preview needs not be a preview but could instead be the actual results.

We adopted this approach together with a slider for the threshold parameter. In this *rapid fine-grained interaction*, spreading is constructed as an incremental and collaborative effort between user and system, rather than being computed after every coarse-grained step, as shown in Figure 9.

Combined these changes allows the user to work both faster and more accurately, albeit while employing more mouse strokes. The more fine-grained interaction lets the user gradually develop a feel for the appropriate threshold, the underlying algorithm, and its limitations by observing many predictions over time.

When evaluated, users perceived that they received results instantaneously (the average response time of the system was 40 ms) and the changes were overall experienced as positive. However, we observed that users tended not to change the threshold parameter at all and leave it at the default – a conservative setting favoring manual control.

Phase 4 – Achieving tight coupling

We had previously observed that some of the annotating pathologists preferred working with a stylus directly on a touch-sensitive screen, instead of a computer mouse with an associated monitor. This preference might be due to the additional eye–hand coordination needed when operating a mouse. Directly placing the pen upon the visual detail to be annotated might be more similar to the prior experience of most users.

For our final prototype, we implemented support for a pressure-sensitive stylus and mapped the user's pressure to the parameter for spreading, which we hoped would let the user learn to control this setting intuitively by the rapid feedback cycle developed in the prior phase. While we were not able to assess this quantitatively, feedback from qualitative evaluations was positive.

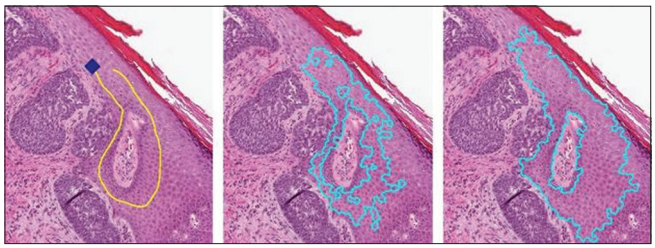

Figure 7: Rejected design: draw and release. A user draws a scribble (left) and upon releasing the mouse button, waits a few moments, and receives the segmentation result (middle *and* right). The results depend on a sensitivity setting, here shown for a conservative (middle) and aggressive setting (right)

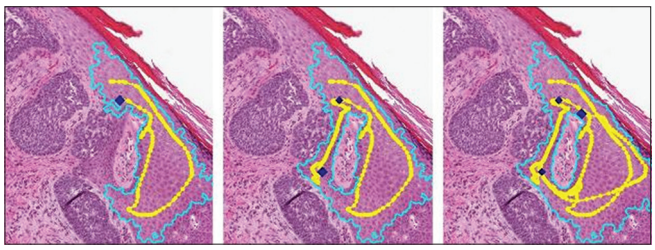

Figure 9: Accepted design: real-time drag and paint. In the final TissueWand version, segmentation updates are received in real time upon click and drag. Points along the path where the user received results are shown as yellow circles along the mouse path (yellow). The resulting segmentation is shown in cyan for three sample points (blue diamonds) along the mouse path (left, middle, right)

User study results

The quantitative user study was carried out on the version of the prototype at the end of Phase 3, i.e., using a mouse as an input device rather than a stylus. The full per case results comparing annotation using manual polygon drawing and using the prototype are available in Appendix B.

The time spent for annotating each case spanned between 10 and 145 min [Figure 10]. On average, the participant performing manual work spent 68 min per case, whereas the two participants using TissueWand spent 38 and 52 min, respectively.

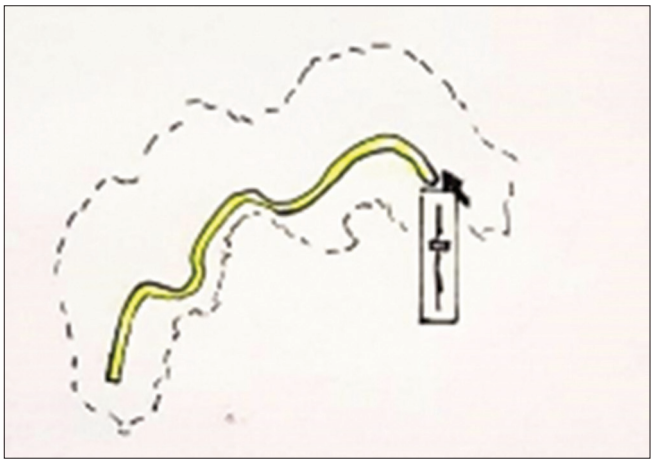

Figure 8: Rejected design: draw and use slider. A user draws a scribble (yellow) and then receives a slider controlling the aggressiveness of automatic segmentation, with live preview results (dashed line)

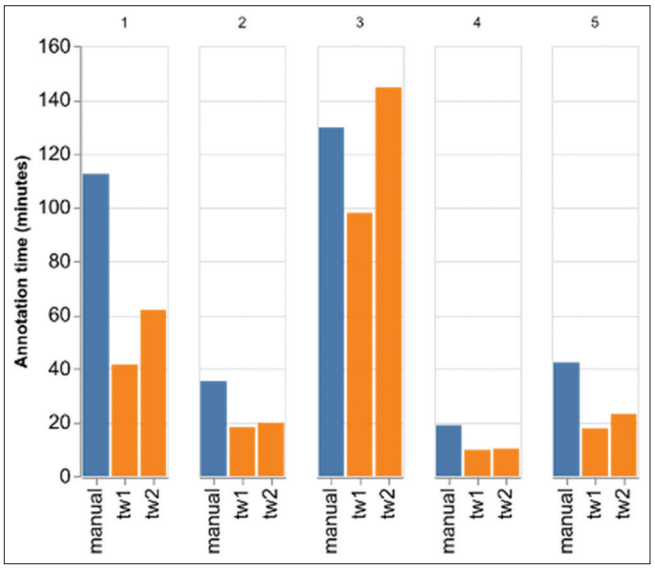

Figure 10: Annotation time per case. Manual, polygon-based annotation time is shown in blue and TissueWand annotation time for two users (tw1, tw2) is shown in orange. The total time required for the five cases spanned between 10 and 145 minutes. TissueWand was faster in all cases, except for case three for user tw2

For most cases, use of the tool meant a substantial speed-up compared to a manual approach. On average across all five cases, we observed a speedup factor of 1.75 corresponding to using 57% of the time required for manual annotation [Figure 11]. This held with one exception. In one single case, TissueWand was both slower and resulted in poorer quality compared to manual annotation.

Quality of annotations between the tools was similar, with a tendency for TissueWand-annotations to be of slightly better quality [Figure 12].

The average error, measured as the distance to the true boundary, was 5.5 μm for manual work, compared to 4.7 and 5.0 μm for the two tool users. A typical error for TissueWand is shown in Figure 13.

The interviews with the participating pathologists after the annotation sessions provide further insights. They concluded on the one hand that the TissueWand tool provided a better experience than manual drawing and that they would choose the tool over manual work for future efforts. On the other hand, a major point was that the annotation effort is still a very demanding process, especially if detailed boundaries are requested. The pathologists felt that the best annotation approach with TissueWand was to start in the interior of a histologic region and work outward toward its boundary.

Discussion

One objective of our work was to design a tool that can effectively assist annotation work. The results indicate that our prototyping effort has been successful. The speed of annotation was higher, the quality was similar or better, and the user preference was for using the tool. Notably, the results are expected to further improve when the stylus support is utilized.

Another aim with the design-based research approach we adopted was to elicit generic knowledge that could be informative for other development efforts with similar characteristics. A key finding during the iterative design process was that rapid response in a local region is a useful principle for annotation support in histopathology. Importantly, the tool does not rely on neither task-specific preprocessing nor specific zoom levels and should therefore be suited for generalization across many histology applications. Another lesson learned from the design phases is that the preferred solution constitutes a tight, dynamic interplay between the underlying real-time processing and the interaction components, such as the display and the input devices.

There are improvement tracks to consider for future evolution of the tool. Some insights are given by Case 3, where the tool benefits were less pronounced. The annotation task here was very fragmented, consisting of outlining many quite small regions. The spreading feature of the tool was not successful in accurately finding the desired boundary for these fragments, and the work essentially reverted to manual drawing.

The proposed approach is constrained by the superpixels generated in the preprocessing step. It is therefore reassuring that the user study showed maintained accuracy compared to manual work. An interesting comparison can be made with FluidAnnotation,[22] where the user interacts by selecting alternative segments from a pretrained model. This is a more limiting constraint than the TissueWand superpixels, which may explain why FluidAnnotation did not reach the accuracy level of manual work.

The interactive mode of TissueWand when used with a mouse is similar to the wand tool of QuPath,[25] although TissueWand allows for more automatic aid and thus further spreading.

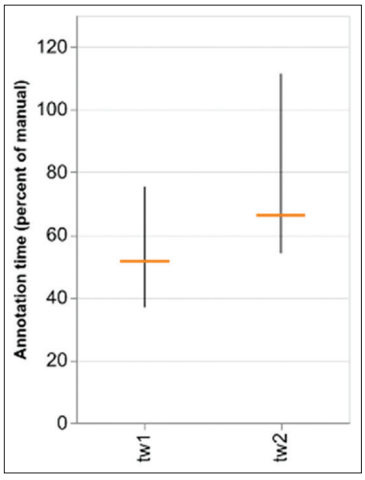

Figure 11: Annotation time relative to manual annotation. Proportion of the manual time for two users (tw1, tw2) over five cases. The vertical bar span minimum and maximum time across cases. The horizontal tick shows the average time. Quantile boxes are omitted due to the low number of observations

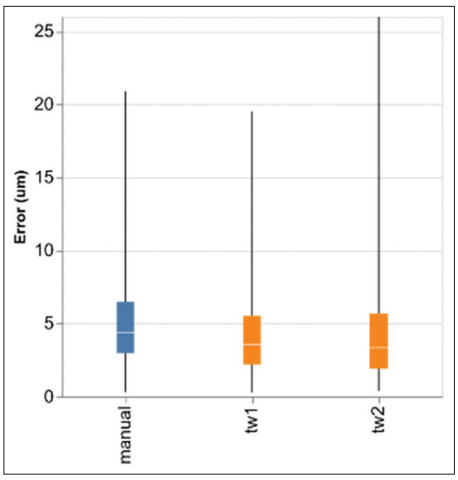

Figure 12: Annotation quality. Error in micrometer between the intended boundary and the drawn annotation. The span between minimum and maximum values is shown with a black vertical line, the first and third quantile are shown as a colored box, and the median is shown as a white line. Values are shown for manual annotation (blue) and two TissueWand users (tw1, tw2 in orange)

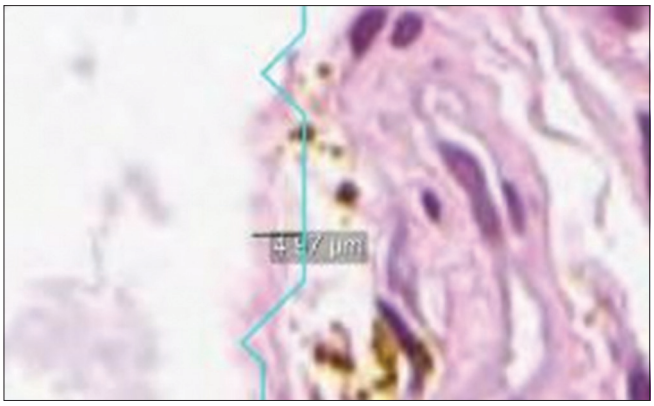

Figure 13: Measuring annotation error. Sample of an error made with TissueWand. Image showing a typical 4.7 μm error made by a participant using the TissueWand tool in ×160. The user's annotation is in cyan, and the error in terms of distance to the intended boundary is shown by a green line

While the predefined generic superpixel boundaries have worked well in our examples so far, there may be annotation tasks where they are suboptimal.

A particularly interesting opportunity for future work is to incorporate a supervised learning component. A classifier such as the one incorporated in TissueWand can be further trained as it is used. Classifier improvements can be used in the interaction by gradually increasing the spread radius and becoming more selective. While more work is needed, our intuition is that the current interaction technique of TissueWand would scale to more capable assistive artificial intelligence.

A limitation of the work presented is of course the small scale of the user study, and future work includes further investigation of the tool's effectiveness across larger data collections, more users, and more applications.

Conclusions

We have presented TissueWand – a rapid interactive tool for assisting annotation efforts. The tool was designed to replace fully manual methods, intended for early stages of dataset curation where no task-specific ML model yet exists to aid the effort. A central characteristic for an assisted annotation tool is how the work is divided between machine and human user over time. One approach would be to let the machine produce a full segmentation and direct the user work to make corrections. Our experiences indicate that this is a problematic strategy for digital pathology, as the cost of finding and correcting errors in the large gigapixel images trumps the gains of the initial automation. Instead, we argue that an interactive tool giving rapid feedback and having high precision in every small interactive step is an effective approach in this context.

Acknowledgments

We would like to thank pathologist Gordan Maras for providing valuable input on the usability of the tool when applied to lymph node metastases.

Financial support and sponsorship

The study was partially supported by Visual Sweden and by the Wallenberg AI, Autonomous Systems and Software Program (WASP).

Conflicts of interest

Martin Lindvall and Claes Lundström are employed by Sectra AB.

References

1. Campanella G, Hanna MG, Geneslaw L, Miraflor A, Werneck Krauss Silva V, Busam KJ, *et al*. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. Nat Med 2019;25:1301-9.
2. Ehteshami Bejnordi B, Veta M, Johannes van Diest P, van Ginneken B, Karssemeijer N, Litjens G, *et al*. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. JAMA 2017;318:2199-210.
3. Litjens G, Kooi T, Bejnordi BE, Setio AA, Ciompi F, Ghafoorian M, *et al*. A survey on deep learning in medical image analysis. Med Image Anal 2017;42:60-88.
4. Lindman K, Rose JF, Lindvall M, Lundström C, Treanor D. Annotations, ontologies, and whole slide images – Development of an annotated ontology-driven whole slide image library of normal and abnormal human tissue. J Pathol Inform 2019;10:22.
5. McGuinness K, O'Connor NE. A comparative evaluation of interactive segmentation algorithms. Pattern Recognit 2010;43:434-44.
6. Mortensen EN, Barrett WA. Intelligent scissors for image composition. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '95. New York, USA: ACM Press; 1995. p. 191-8.
7. Falcão AX, Udupa JK, Samarasekera S, Sharma S, Hirsch BE, de Lotufo RA. User-steered image segmentation paradigms: Live wire and live lane. Graph Model Image Process 1998;60:233-60.
8. Boykov YY, Jolly MP. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proceedings Eighth IEEE International Conference on Computer Vision ICCV. IEEE Computer Society; 2001. p. 105-12.
9. Mathieu B, Crouzil A, Puel JB. Interactive segmentation: A scalable superpixel-based method. J Electron Imaging 2017;26:61606.
10. Rother C, Kolmogorov V, Blake A, GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts, in ACM SIGGRAPH 2004 Papers. New York, NY, USA: ACM Press; 2004, p. 309–14.
11. Malmberg F, Nordenskjöld R, Strand R, Kullberg J. SmartPaint: a tool for interactive segmentation of medical volume images. Comput Methods Biomech Biomed Eng Imaging Vis 2014;5:36-44.
12. Fails JA, Olsen DR. Interactive machine learning. In: Proceedings of the 8th International Conference on Intelligent user Interfaces - IUI '03. New York, USA: ACM Press; 2003. p. 39.
13. Dudley JJ, Kristensson PO. A review of user interface design for interactive machine learning. ACM Trans Interact Intell Syst 2018;8:1-37.
14. Amershi S, Cakmak M, Knox WB, Kulesza T. Power to the people: The role of humans in interactive machine learning. AI Mag 2014;35:105-20.
15. Arendt D, Saldanha E, Wesslen R, Volkova S, Dou W. Towards Rapid Interactive Machine Learning: Evaluating Tradeoffs of Classification Without Representation. In: Proceedings of the 24th International Conference on Intelligent User Interfaces. New York, NY, USA: ACM; 2019. p. 591-602.
16. Settles B. Active Learning Literature Survey. University of Wisconsin-Madison [Internet]. 2009 Jan [cited 2019 Sep 24]. Available from: https://minds.wisconsin.edu/handle/1793/60660.
17. Dietterich TG, Lathrop RH, Lozano-Pérez T. Solving the multiple instance problem with axis-parallel rectangles. Artif Intell 1997;89:31-71.
18. Yang L, Zhang Y, Zhao Z, Zheng H, Liang P, Ying MT, *et al*. BoxNet: Deep Learning Based Biomedical Image Segmentation Using Boxes

Only Annotation. arXiv180600593 [cs]; 2018.
19. Lin D, Dai J, Jia J, He K, Sun J. ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation. arXiv160405144 [cs]; 2016.
20. Castrejon L, Kundu K, Urtasun R, Fidler S. Annotating Object Instances with a Polygon-RNN. arXiv170405548 [cs]; 2017.
21. Acuna D, Ling H, Kar A, Fidler S. Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. arXiv180309693 [cs]; 2018.
22. Andriluka M, Uijlings JR, Ferrari V. Fluid Annotation: A Human-Machine Collaboration Interface for Full Image Annotation; 2018.
23. Tizhoosh HR, Pantanowitz L. Artificial intelligence and digital pathology: Challenges and opportunities. J Pathol Inform 2018;9:38.
24. Rubens U, Hoyoux R, Vanosmael L, Ouras M, Tasset M, Hamilton C, *et al*. Cytomine: Toward an open and collaborative software platform for digital pathology bridged to molecular investigations. Proteomics Clin Appl 2019;13:e1800057.
25. Bankhead P, Loughrey MB, Fernández JA, Dombrowski Y, McArt DG, Dunne PD, *et al*. QuPath: Open source software for digital pathology image analysis. Sci Rep 2017;7:16878.
26. Aubreville M, Bertram C, Klopfleisch R, Maier A. SlideRunner – A Tool for Massive Cell Annotations in Whole Slide Images. arXiv180202347; 2018.
27. Puttapirat P, Zhang H, Lian Y, Wang C, Zhang X, Yao L, *et al*. OpenHI - An open source framework for annotating histopathological image. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM); 2018. p. 1076-82.
28. Ewing L, Ahn SW, Jonas OH, Hata N. Pixelwise tissue segmentation for precise local in-vivo dose response assessment in patient-derived xenografts. In: Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling. International Society for Optics and Photonics; 2019. p. 1095133.
29. Homeyer A, Schenk A, Arlt J, Dahmen U, Dirsch O, Hahn HK. Practical quantification of necrosis in histological whole-slide images. Comput Med Imaging Graph 2013;37:313-22.
30. Nalisnik M, Gutman DA, Kong J, Cooper LA. An interactive learning framework for scalable classification of pathology images. Proc IEEE Int Conf Big Data 2015;2015:928-35.
31. Lutnick B, Ginley B, Govind D, McGarry SD, LaViolette PS, Yacoub R, *et al*. An integrated iterative annotation technique for easing neural network training in medical image analysis. Nat Mach Intell 2019;1:112-9.
32. Bulten W, Bándi P, Hoven J, Loo RV, Lotz J, Weiss N, *et al*. Epithelium segmentation using deep learning in H & E-stained prostate specimens with immunohistochemistry as reference standard. Sci Rep 2019;9:864.
33. Burlutskiy N, Pinchaud N, Gu F, Hägg D, Andersson M, Björk L, *et al*. Segmenting Potentially Cancerous Areas in Prostate Biopsies using Semi-Automatically Annotated Data; 2019.
34. Sommer C, Straehle C, Köthe U, Hamprecht FA. Ilastik: Interactive learning and segmentation toolkit. In: 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro; 2011. p. 230-3.
35. Sanner A, Petré F. Semi-Automated Annotation of Histology Images: Development and Evaluation of a user Friendly Toolbox. Linköping University; 2016.
36. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 2012;34:2274-82.
37. Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 2002;24:971-87.
38. Lucchi A, Smith K, Achanta R, Lepetit V, Fua P. A fully automated approach to segmentation of irregularly shaped cellular structures in EM images. Med Image Comput Comput Assist Interv 2010;13:463-71.
39. Smith K, Carleton A, Lepetit V. Fast ray features for learning irregular shapes. In: 2009 IEEE 12th International Conference on Computer Vision. IEEE; 2009. p. 397-404.

Appendixes

Appendix A: Superpixel flood fill pseudocode

Pseudocode: flood fill affinity search

```
Input: threshold, hit_superpixels
MAX_DISTANCE=9 # number of superpixel steps
feature_hit = median(features[hit_superpixels])
seen_before = set([])
matched = set([hit_superpixels])
candidates = neighbours(hit_superpixels) - seen_before
while len(candidates) >0:
new_candidates = set()
for spx in candidates:
distance = minimum_path(hit_superpixels)
similarity = cosine_similarity(feature_hit, features[spx])
if similarity > threshold + distance*((1.0-threshold)/MAX_
DISTANCE):
new_candidates |= spx
matched |= new_candidates
candidates = new_candidates - seen_before
```

Appendix B: Quantitative user study results

Table 1: The speed and quality measures are given for the annotations across the five cases

Case	Annotation time (min)			Speed-up factor		Annotation quality (μm error)		
	Manual	TW1	TW2	TW1	TW2	Manual	TW1	TW2
1	112	42	62	2.7	1.8	6.5	5.9	4.0
2	35	18	20	1.9	1.8	5.7	4.5	5.2
3	130	98	145	1.3	0.9	5.2	5.3	6.0
4	19	10	10	1.9	1.9	4.1	3.0	4.1
5	42	18	23	2.4	1.8	6.2	4.7	5.6
Average	68	37	52	2.0	1.6	5.5	4.7	4.9

TW1 and TW2 refer to the users 1 and 2 employing the TissueWand tool. For clarity, the speed-up factor compared to the manual condition is also reported separately. Overall, the tool causes substantial reductions in annotation time, without loss of annotation quality