

FlexTileSource: An OpenSeadragon Extension for Efficient Whole-Slide Image Visualization

Peter J. Schüffler^{1,2}, Gamze Gokturk Ozcan¹, Hikmat Al-Ahmadie¹, Thomas J. Fuchs^{1,3}

¹Department of Pathology, Memorial Sloan Kettering Cancer Center, New York City, NY, USA, ²TUM School of Medicine, Technical University of Munich, Munich, Germany, ³Department of Pathology, Icahn School of Medicine at Mount Sinai, New York, New York, USA

Submitted: 19-Feb-2021

Revised: 03-May-2021

Accepted: 01-Jul-2021

Published: 14-Sep-2021

Abstract

Background: Web-based digital slide viewers for pathology commonly use OpenSlide and OpenSeadragon (OSD) to access, visualize, and navigate whole-slide images (WSI). Their standard settings represent WSI as deep zoom images (DZI), a generic image pyramid structure that differs from the proprietary pyramid structure in the WSI files. The transformation from WSI to DZI is an additional, time-consuming step when rendering digital slides in the viewer, and inefficiency of digital slide viewers is a major criticism for digital pathology. **Aims:** To increase efficiency of digital slide visualization by serving tiles directly from the native WSI pyramid, making the transformation from WSI to DZI obsolete. **Methods:** We implemented a new flexible tile source for OSD that accepts arbitrary native pyramid structures instead of DZI levels. We measured its performance on a data set of 8104 WSI reviewed by 207 pathologists over 40 days in a web-based digital slide viewer used for routine diagnostics. **Results:** The new *FlexTileSource* accelerates the display of a field of view in general by 67 ms and even by 117 ms if the block size of the WSI and the tile size of the viewer is increased to 1024 px. We provide the code of our open-source library freely on <https://github.com/schuefflerlab/openseadragon>. **Conclusions:** This is the first study to quantify visualization performance on a web-based slide viewer at scale, taking block size and tile size of digital slides into account. Quantifying performance will enable to compare and improve web-based viewers and therewith facilitate the adoption of digital pathology.

Keywords: Digital pathology, OpenSeadragon, viewer optimization, whole-slide imaging

INTRODUCTION

The visualization of whole-slide images (WSI) on a computer screen is a central part in digital pathology. WSI is high-resolution scans of glass slides, often scanned at 0.5 $\mu\text{m}/\text{pixel}$ (mpp) or 0.25 mpp which corresponds to optical magnifications of $\times 20$ or $\times 40$, respectively. Higher resolutions are also possible but less common. Depending on the resolution and the tissue amount on the slide, the resulting WSI can be of gigapixel size with dimensions of $100,000 \times 100,000$ px or more. No computer screen exists which can display a WSI as a whole. For example, a standard screen of 1900×1080 px can display only 0.02% of the above dimensions.

Therefore, many web-based digital slide viewers^[1-7] are built using OpenSlide and OpenSeadragon^[8] (OSD), an open-source library to view and navigate high-resolution images (there are also other techniques, such as OpenLayers^[9] or own developments, e.g., Bio-Atlas^[10] or Brain Maps^[11]). Specifically, OSD provides the class *DziTileSource* to interact with deep zoom images (DZI). The DZI format is a specific

structure currently described and maintained by Microsoft^[12] to store large images in a pyramidal way. The basis of the pyramid forms the original image that is then downscaled by a factor of two to generate the consecutive level. This level is then again downscaled by a factor of two to build the next level, and so forth, until the resulting level has a dimension of 1×1 px (half pixels by division of odd dimensions are rounded up). Further, every level is cut in separate squared blocks with fixed size. This increases the total file size of the image but comes with several advantages for image streaming in a viewer.

Address for correspondence: Prof. Dr. Peter J. Schüffler,
Technical University of Munich, Munich, Germany,
New York City, NY, USA.
E-mail: peter.schueffler@tum.de

This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License, which allows others to remix, tweak, and build upon the work non-commercially, as long as appropriate credit is given and the new creations are licensed under the identical terms.

For reprints contact: WKHLRPMedknow_reprints@wolterskluwer.com

How to cite this article: Schüffler PJ, Ozcan GG, Al-Ahmadie H, Fuchs TJ. Flextilesource: An openseadragon extension for efficient whole-slide image visualization. *J Pathol Inform* 2021;12:31.

Available FREE in open access from: <http://www.jpathinformatics.org/text.asp?2021/12/1/34/326169>

Access this article online

Quick Response Code:



Website:
www.jpathinformatics.org

DOI:
10.4103/jpi.jpi_13_21

In a web viewer, rather than downloading the complete DZI file, OSD displays only the currently viewed part of the DZI by querying image tiles around the current location and zoom level on the image. The tile size in the viewer can be set separately and does not necessarily correspond to the block size in the file. OSD further allows the user to zoom and navigate through the image. The DZI pyramid enables OSD to sample the current subimage efficiently from higher pyramid levels, which is particularly useful when the user zooms out of the image.

However, even though WSI is scanned in a pyramidal format, they do not necessarily comply with the DZI standard. For example, with the scanner AT2 (Leica Biosystems, Buffalo Grove, Illinois, USA), WSI is scanned with 2–4 levels only, where every level is a downscale of the factor 4 of the one below. Since OSD’s *DziTileSource* expects a standard DZI as input, a translation has to be made from the proprietary image pyramid to DZI [Figure 1].

This translation is conducted by the *DeepZoomGenerator* extension of the open-source library OpenSlide^[13] (OS). While OS provides libraries to open and read tiles from WSI of various scanner vendors, *DeepZoomGenerator* generates a virtual DZI structure around the native image pyramid of the digital slide. It does so by identifying the best suitable native pyramid level for each DZI level as the closest level with equal or larger dimension and downscaling blocks from this native level to the DZI level resolution. This conversion can cost time in situations where the WSI does not contain all levels of a DZI.

For example, if a tile from a DZI level is queried by OSD and there is a corresponding native level in the WSI, the tile can directly be taken from that native level and no downscaling is needed. If a tile from a DZI level is queried in between two

native levels of the WSI, OS will adjust the tile coordinates to the next larger native level, retrieve the larger tile there, and downscale it back to the queried resolution. Depending on the level mismatch of the proprietary WSI and the DZI, this tile conversion can happen many times during a slide review and add a certain delay to the viewing experience. Note that, OS employs a comprehensive caching strategy such that downscaled tiles do not have to be recreated again when revisiting the same area. However, in routine slide review, most parts of a slide are inspected only once such that the “first-time” view, in which OS cannot benefit from the cache, is of most importance.

In this work, we provide a new tile source definition *FlexTileSource*, as an alternative to OSDs *DziTileSource*. *FlexTileSource* is flexible in the specification of the image pyramid and does not require a DZI format. One can specify the number, dimensions, and tile sizes of the pyramid levels, and OSD will only query for tiles from the given levels instead of querying for tiles from the standard DZI pyramid. This makes internal steps of tile downscaling and additional queries for tiles on intermediate levels obsolete, thereby increasing the speed at which tiles can be provided to the viewer’s field of view (FOV), ultimately increasing the overall performance.

FlexTileSource has been tested over 40 days using an in-house developed web-based slide viewer used by 207 pathologists in a routine diagnostic workflow.^[14,15,16] We show that it consistently increases the visualization performance by 50 ms/FOV, with progressive efficiency on WSI with more native levels. Further, we show that FOVs are served faster with larger block and tile sizes than used today. *FlexTileSource* is applicable not only to WSI but also to any other zoomable image that has a native non-DZI standard pyramid structure.

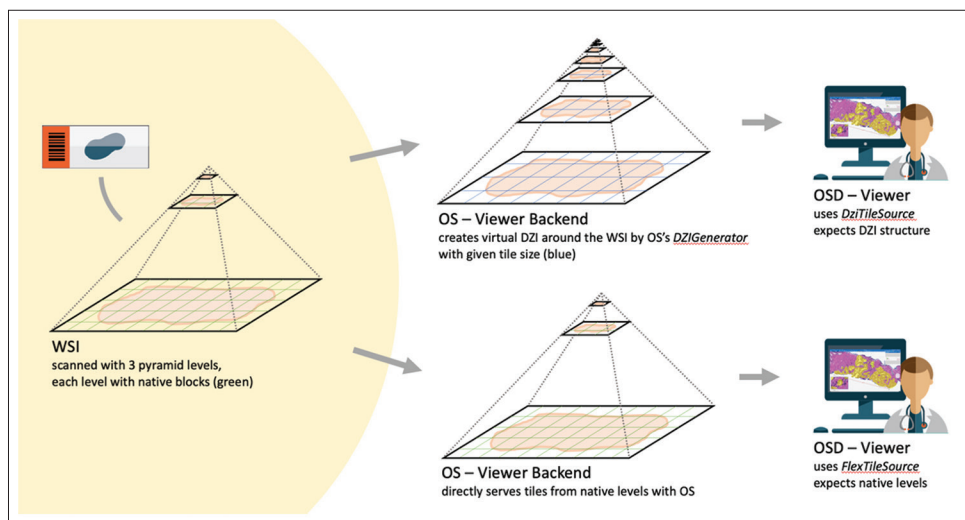


Figure 1: Two different tile serving approaches. With the traditional OpenSlide *DziTileSource* (top), the native image pyramid of a whole-slide image (left) with fixed block size and arbitrary levels has to be “converted” to a deep zoom image format with all pyramid levels and viewer specific tile size. Depending on the displayed field of view, stitching, cropping, and downscaling of blocks are necessary to produce a tile. Alternatively, we propose *FlexTileSource*, which directly picks tiles from the whole-slide image pyramid levels (bottom). Repetitive access to intermediate levels is not needed. Instead, the viewer uses the tiles longer before “jumping” to another level, removing the need to load and generate tiles from nonexistent levels

The library is open source, free to use, and available at <https://github.com/schuefflerlab/openseadragon>.

Besides DZI, OSD provides tile sources for Legacy Image Pyramids (International Image Interoperability Framework [IIIF]), (open street maps [OSMs]), (tiled map service [TMS]), Zoomify, Custom Tile Sources, and Simple Image. Legacy Image Pyramids are intended for pyramids of full images rather than tiles and thus are not suitable to high-resolution images. IIIF is a quite flexible image standard but only supports full integer downscales of pyramid levels. OSM, TMS, Zoomify, and custom Tile Sources assume “perfect” pyramids by halving levels, similar as DZI, and have been developed for cartographic maps of geo-referenced data. Simple Image does not employ zoomable high-resolution images at all. Therefore, we propose *FlexTileSource* as a complementary extension of the powerful set of existing tile sources, tailored to the particularities of WSI.


METHODS

To be compatible with OSD, the new tile source is described by an XML document with the extension *flex*. It specifies the new image type *flex-image-pyramid*, the file format of the tiles, and all levels of the pyramid by their width, height, tile width, and tile height [Figure 2]. Internally, *FlexTileSource* is built as a new OSD class that implements the required interfaces of a tile source, namely *supports*, *configure*, and *getTileUrl*. *supports* detects whether a new image file is valid and supports the new image type *flex-image-pyramid*. *Configure* is based on the implementation of the *DziTileSource* interface but has been adjusted to read the *flex* files as custom definitions of pyramid levels instead of calculating pyramid levels based on the base image only. *getTileUrl* has been implemented to query tiles by their slide level and address, i.e., https://tileservers/{level}/{x}_{y}.{fileformat}.

Where *level* is the native pyramid level number (level 0 being the base image,) and *x* and *y* are the coordinates of the queried tile on that level assuming a tile width and height as specified in the xml. For a detailed description of the tile source, we refer to the source code on <https://github.com/schuefflerlab/openseadragon>.

Data and experiment

We included the new tile source in a web-based digital slide viewer that is routinely used for clinical purposes. Over a time period of 40 days, 207 distinct pathologists accessed 8104 WSI



Example Query:
`https://imageserver/wsi.flex`

Example Result:

```
<image type="flex-image-pyramid" fileFormat="jpg" capture-installed="true">
  <level width="61917" height="43923" tileWidth="1024" tileHeight="1024"/>
  <level width="15479" height="10980" tileWidth="1024" tileHeight="1024"/>
  <level width="3869" height="2745" tileWidth="1024" tileHeight="1024"/>
  <level width="1024" height="726" tileWidth="1024" tileHeight="726" />
  ...
</image>
```

Figure 2: Example query and response for a flex image pyramid description of a whole-slide image with 4 levels and original width and height of 61,917 × 43,923 px

that meets all three inclusion criteria. First, WSI were included that was digitized with an AT2 scanner (Leica Biosystems, Buffalo Grove, IL, USA), as their native pyramid information can be easily be read and their block size can be configured on the scanner level. Second, WSI were included that have been opened from the laboratory information system to focus on slide reviews in a clinical setting. Although we do not expect major differences to other settings (research, education), where slides might be reviewed for a longer time, repetitive, or be annotated, we want to exclude the possibility of any variations in this setting as much as possible. Furthermore, in a clinical setting, WSI are typically opened the first time such that caching artifacts are largely excluded. Finally, included WSI had to be JPEG compressed to exclude compression and decompression influences.

3214 of the WSI were internally opened using the proposed *FlexTileSource* and 4890 with OSD’s *DziTileSource* as comparison. To investigate the relationship between block size, tile size, and the two tile sources, we applied different block sizes to the scanner settings prior scanning (240 px, 256 px, 512 px, and 1024 px) and different tile sizes in the viewer (512 px, 1024 px, and 1980 px).

During their slide review, pathologists navigated through the slide including zooming and panning. For every navigation event, OSD will load new tiles to render the target FOV. Both, the navigation and the complete loading of a FOV are programmatically detectable events in OSD. These events were used to record the time taken from the beginning of a navigation to the completion of the FOV, together with the number of loaded tiles and the size of the FOV. Note that, if the user continues moving through the image without waiting for the FOV to be fully loaded, the tile counter and the time to render the FOV will continue to grow. To avoid side effects in such scenarios such as networking queue effects and other latencies, we included only events in which the user did not continue to move the FOV before it was fully loaded, which happened for 56,440 FOVs.

The raw time to display the FOV T_{FOV} in ms was normalized to a standard FOV size of 1920 × 1080 px to compensate for different window and monitor sizes and for different tile sizes. This is done by multiplying T_{FOV} by the factor of expected versus actual number of tiles n : $T_{sFOV} = T_{FOV} \times (n_{exp}/n)$,

Where n_{exp} is the expected number of tiles for the standard FOV size.

Every tile created and sent to the client was also cached in the viewer backend such that the FOV was loaded faster when the user came back to the same location. The time to render a FOV will also depend on the network speed and viewer server specifications. All our experiments were conducted with a 1 Gb/s network between viewer server and with a viewer server as specified earlier.^[14]

RESULTS

Pathologists launched the viewer from the LIS in a web browser with a median FOV width of 1920 px (360 px– 5120 px) and height of 887 px (271 px– 2260 px).

Figure 3 illustrates the overall time differences for FOVs of slides that were internally visualized using the *FlexTileSource* or *DziTileSource*. The median time to fully load the FOV was 308 ms or 375 ms, respectively. Our proposed tile source was significantly 67 ms faster (student's *t*-test $P < 0.01$). The only difference of the two scenarios is the use of a tile source which queries tiles only from native slide levels instead of additional intermediate levels.

We illustrate the impact of the number of native levels in Figure 4. Slides with 2 native levels don't profit significantly from the new tile source ($P = 0.974$), while slides with 3 or 4 native levels increasingly benefit with 51 ms ($P < 0.01$) and 75 ms ($P < 0.01$) median performance gain.

Block size and tile size

An important relationship exists between block size and tile size. *Blocks* are static image patches with fixed size saved in the WSI during scanning whereas *tiles* are image patches with configurable size displayed in the viewer. There can be three scenarios how tiles are formed from the blocks: first, if the tiles in the viewer are of the same size as the blocks in the WSI, the tile server can serve the blocks as they are. Second, if the tile size is a multiple of the block size, blocks will be stitched together to form a tile. Third, if the tiles are of unrelated size to each other, the tile server stitches and crops blocks to form the resulting tile. In our experiment, the three scenarios resulted in varying FOV loading time in the different scenarios. We utilized WSI with block sizes of either 240 px, 256 px, 512 px, or 1024 px, and tile sizes of either 512 px, 1024 px or 1920 px, respectively. Figure 5 shows the FOV loading time of the different block and tile sizes as measured by the viewer. We grouped WSI with equal tile sizes and equal block

sizes together and compared the loading time of our new tile source with OSD's DZI tile source. The first group of 8 boxes comprises WSI with a tile size of 512 px. Half of them have been served with the original *DziTileSource* (red) and half of them with the new *FlexTileSource* (blue). Further, a quarter of those WSI used a block size of 240 px, 256 px, 512 px, or 1024 px, respectively. The next 8 boxes contain only WSI with a viewer tile size of 1024 px and so on. Four insights are remarkable here: first, with a tile size of 512 px, the FOV needs at least 155 ms to be fully rendered, regardless of the block size. With larger tiles, the FOV can be displayed already after 26 ms. This can be explained by the higher number of tiles needed for the FOV: at 512 px, on average 12 tiles needed to fit a 1920 × 1080 px FOV, while for 1024 px and 1920 px tiles, only 4 tiles are needed.

Second, larger WSI block sizes lead to faster FOV rendering regardless of the tile size. This can be explained by that large blocks do not require stitching very often (but mostly cropping). A large block can then be reused by OS to generate the neighboring tile, without the need to read the neighbor block from the WSI file.

Third, constellations in which the tile size is not a multiple of the block size or vice versa result the highest variance. This could indicate the higher number of block stitching and cropping processes which require the longest time to generate a tile altogether.

These three observations depend on block, tile, and FOV size only, and not on the tile source. However, the fourth observation illustrates the performance advantage of the *FlexTileSource*: the overall fastest configuration to display a full FOV was at a block and tile size of both 1024 px using the

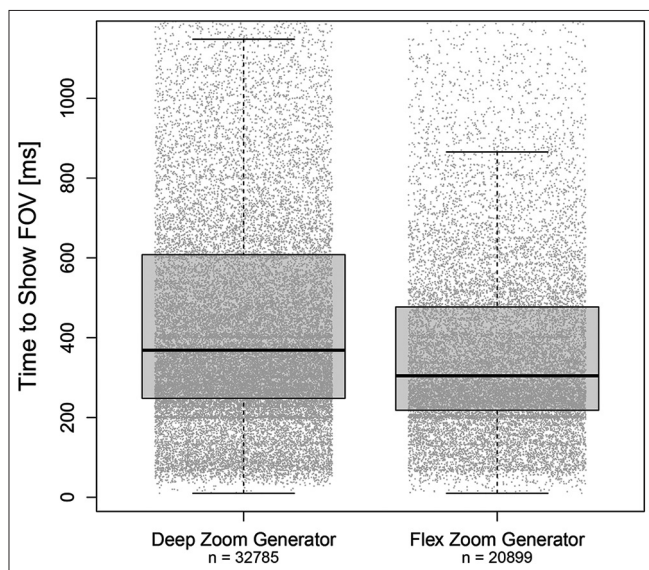


Figure 3: Overall performance gain to display a field of view is 67 ms ($P < 0.01$). Left: 32,785 field of view change events using the state-of-the-art *DziTileSource* took 375 ms to fully load the field of view. Right: 20,899 field of view change events using the new *FlexTileSource* needed 308 ms. This includes all block sizes and tile sizes

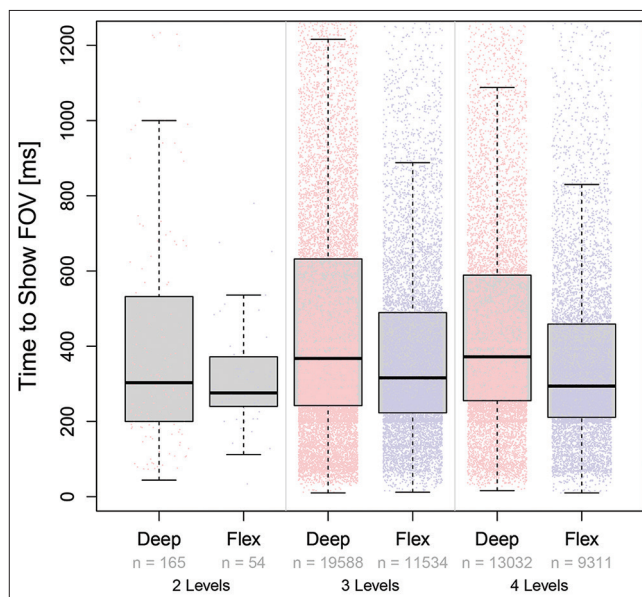


Figure 4: The benefit of the new *FlexTileSource* is larger the more levels the whole-slide image has. For 2-level whole-slide image, Δt is 18 ms ($P = 0.974$); for 3-level whole-slide image, Δt is 51 ms ($P < 0.01$); and for 4-level whole-slide image, Δt is 75 ms ($P < 0.01$)

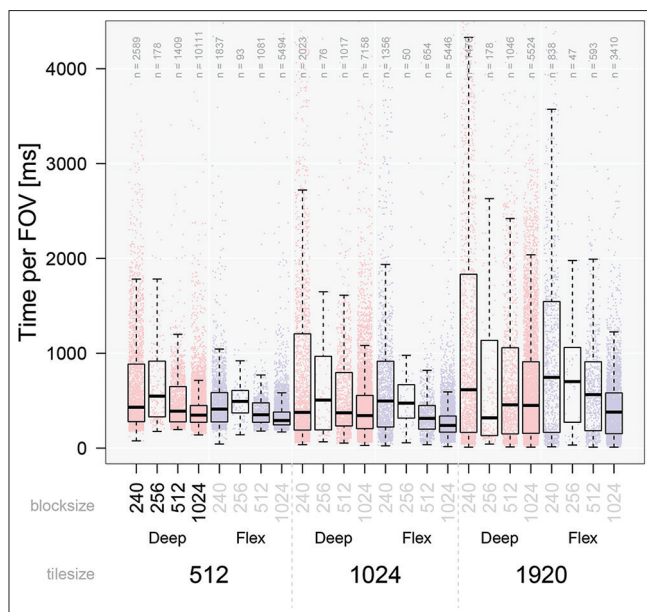


Figure 5: Normalized field of view loading time in dependency of whole-slide image block size and viewer tile size. Red – whole-slide image using OpenSeadragon’s *DziTileSource*, Blue – whole-slide image using the new *FlexTileSource*. Insight 1: The larger the block size in the file, the faster the field of view can be served. Insight 2: whole-slide image using *FlexTileSource* can be served faster than with *DziTileSource*. Insight 3: The fastest constellation is a block size and tile size of 1024 px together with the new *FlexTileSource* (median 240 ms)

new tile source only. With a median time of 313 ms per FOV, this is 117 ms faster ($P < 0.01$) than the fastest configuration with the traditional tile source at a block size of 240 px and a tile size of 512 px (430 ms).

DISCUSSION AND CONCLUSION

Web-based digital slide viewers that use OSD and OS commonly mimic a DZI structure around the WSI for visualization. Creating tiles on virtual levels which are not present in the slide file can cost time for stitching, cropping, downscaling, and transfer over the network.

Therefore, we propose a new tile source which directly streams tiles from the native levels of a slide instead of from the DZI format. The native image pyramid can comprise an arbitrary number of levels with arbitrary dimensions, and no regular or systematic pyramid structure is needed. Furthermore, every level can potentially use arbitrary tile sizes. This makes our tile source flexible and applicable to zoomable images other than WSI.

We showed on a dataset of 8104 WSI reviewed by 207 pathologists in a production clinical setting that our tile source is able to significantly increase the speed to display a FOV overall by 67 ms. Further, we showed that the *FlexTileSource* is faster the more native levels a WSI has.

We illustrated the relationship of a WSI’s block size and the viewer’s tile size. Larger block sizes tend to be faster for the viewing experience since OS has to access the file less often

than with smaller blocks and since fewer stitching operations have to be made. In general, block size and tile size should be proportional to avoid too many stitching and cropping processes. We demonstrated that a block size and a tile size of 1024 px each, together with the new tile source, led to significantly fastest viewing experience with a rendering time of 313 ms/FOV for a modern computer screen, 117 ms faster than standard approaches with a block size of 256 px and tile size of 512 px that are widely used today (430 ms).

The impact of our tile source on the efficiency for digital sign out on a case level needs yet to be assessed in future research. Still, there is no doubt that fast and seamless web viewers will facilitate the adoption of digital pathology. And with our study, we provide another step into this direction with a simple software extension to tune the time to render a FOV. This can further be useful when additional tile postprocessing is applied such as color alterations, on devices with slow network connection, or in other cases where the overall FOV rendering time is valuable. Finally, while this study focuses on improvements for web-viewers that use OS and OSD, there are further streaming improvements possible when using different technologies in the first place, such as pretiling or other image formats, which are to be investigated in future studies.

Financial support and sponsorship

This research was funded in part through the NIH/NCI Cancer Center Support Grant P30 CA008748.

Conflicts of interest

PJS and TJF are both co-founder of Paige. The authors declare no other conflicts of interest.

REFERENCES

- McLendon R, Friedman A, Bigner D, Van Meir EG, Brat DJ, M. Mastrogiannis G, *et al.* Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* 2008;455:1061-8.
- Virtual Pathology at the University of Leeds. Available from: <https://www.virtualpathology.leeds.ac.uk/>. [Accessed on 2021-Feb-12]
- Digital Slide Archive (DSA). Available from: <https://cancer.digitalslidearchive.org/#!/CDSA/acc/TCGA-OR-A5J1/TCGA-OR-A5J1-01A-01-TS1?slide=TCGA-OR-A5J1-01A-01-TS1.CFE08710-54B8-45B0-86AE-500D6E36D8A5.svs>. [Accessed on 2021-Feb-12]
- Pathpresenter Public. Available from: <https://pathpresenter.net/home>. [Accessed on 2021-Feb-12]
- Objective Whole Slide Image Server. Available from: <https://images.objectivepathology.com/default.aspx>. [Accessed on 2021-Feb-12]
- Pathorama - Welcome. Available from: <https://pathorama.ch/>. [Accessed on 2021-Feb-12]
- Organpathologie-Atlas | el-IPH Organpathologie-Atlas. Available from: <https://eliph.klinikum.uni-heidelberg.de/atlas>. [Accessed on 2021-Feb-12]
- OpenSeadragon-An Open-Source, Web-Based Viewer for High-Resolution Zoomable Images, Implemented in Pure JavaScript, for Desktop and Mobile. Available from: <http://openseadragon.github.io/>. [Accessed on 2021-May-10]
- OpenLayers - Welcome. Available from: <https://openlayers.org/>. [Accessed on 2021-Feb-12]
- Sabalianskas NA, Foutz CA, Mest JR, Budgeon LR, Sidor AT, Gershenson JA, *et al.* High-throughput zebrafish histology. *Methods* 2006;39:246-54.

11. Jones EG, Stone JM, Karten HJ. High-resolution digital brain atlases: A Hubble telescope for the brain. *Ann N Y Acad Sci* 2011;1225:E147-59.
12. Deep Zoom File Format Overview. Available from: [https://docs.microsoft.com/en-us/previous-versions/windows/silverlight/dotnet-windows-silverlight/cc645077\(v=vs.95\)](https://docs.microsoft.com/en-us/previous-versions/windows/silverlight/dotnet-windows-silverlight/cc645077(v=vs.95)). [Accessed on 2021-jan-29]
13. Goode A, Gilbert B, Harkes J, Jukic D, Satyanarayanan M. OpenSlide: A vendor-neutral software foundation for digital pathology. *J Pathol Inform* 2013;4:27.
14. Schüffler PJ, Geneslaw L, Yarlagadda DVK, Hanna MG, Samboy J, Stamelos E, *et al.* Integrated digital pathology at scale: A solution for clinical diagnostics and cancer research at a large academic medical center. *JAMIA* 2021;28(9):1874–1884 doi <https://doi.org/10.1093/jamia/ocab085>
15. Hanna MG, Reuter VE, Hameed MR, Tan LK, Chiang S, Sigel C, *et al.* Whole slide imaging equivalency and efficiency study: experience at a large academic center. *Mod Path* 2019;32:916-928 doi 10.1038/s41379-019-0205-0
16. Hanna MG, Reuter VE, Ardon O, Kim D, Sirintrapun SJ, Schüffler PJ, *et al.* Validation of a digital pathology system including remote review during the COVID-19 pandemic. *Mod Path* 2020;33:2115-2127 doi 10.1038/s41379-020-0601-5