DOI: 10.2352/issn.2168-3204.2016.1.0.95

The "International Image Interoperability Framework" and its Implication to Preservation

Lukas Rosenthaler and Peter Fornaro; Digital Humanities Lab; University of Basel; Basel, Switzerland

Abstract

The new "International Image Interoperability Framework" (IIIF)[1] *defines a standardised URL syntax to serve digital images online in the field of cultural heritage and research. The region of interest, resolution, rotation and file format can be indicated on the URL. However, most of the current implementations are not specifically designed for the uses in an environment were persistence and permanence is crucial. A well implemented and embedded IIIF-server has therefore positive implications to the organisation of long-term preservation of digital image files. In the context of a national data center for the long-term access to research data in the humanities, we implemented an IIIF compliant image server with both high performance services for virtual research environments as well as long-term preservation of these images in mind.*

Introduction

Making digital images available over the internet has been done since the advent of Internet. The HTTP protocol offered a simple way to address an image by it's unique URL. Given the proper format – usually JPEG – the image can be displayed directly in a browser, otherwise it has to be downloaded. However, as the capture technology matured, the quality and the size of the master files grew to in such a way that the high quality master files couldn't be made available through this simple technology. In addition, in order to avoid the degradation that lossy file formats such as JPEG impose, the master files are often stored in the "heavy weight" TIFF format possibly using a 16 Bit samples at highest resolution.

With the exception of directly linking an image by an URL and a web server which transfers the image "as is" there have been no means to refer to images. Embedding images in web applications such as virtual research environments therefore requires that many copies in different resolutions and quality levels have to be stored and presented with different URL's for each version. This is both inefficient and cumbersome on the repository side as well as on the side of the application which wants to use the images. This holds especially true if using the "linked open data" (LOD) concepts where the function of the repository ("data provider") and the data user (a LOD application) may belong to different institutions.

Within the repository, managing different versions in addition to the master file is a tedious task. While in theory, long term archiving of the master file should be sufficient, also the derivates for presenting have to be kept save (using complicated backup strategies) since the creation of the derivatives from the master files is a tedious process.

[1] `http://iiif.io`.

The "International Image Interoperability Framework" (IIIF) could offer an elegant solution to both sides of the problem.

International Image Interoperability Framework (IIIF)

The IIIF basically defines a standard how images should be accessed using a http connection and a specially formatted URL (which otherwise conforms to the standard syntax of URl's). Its goal is to enable the sharing of digital images using the Linked Open Data (LOD) principles in the field of cultural heritage images and humanities research. It has been conceived by a large consortium of serious players in the field of cultural heritage and research institutions such as ARTstore, the British Library, Harvard University, to name a few. The IIIF syntax offers the following functions:

- Selection of a region of interest. This allows high resolution viewers depending on tiles and a resolution pyramid to access images served by IIIF.
- Selection of the resolution resp. size of the presented image
- Arbitrary rotation and mirroring
- Image quality (meaning B/W, grey, color)
- File format used to present the image

The IIIF URL-syntax is as follows

```
{scheme}://{server}{/prefix}/{identifier}
/{region}/{size}/{rotation}/{quality}.{format}
```

where

scheme denotes the protocol, either *http* or *https*.

server indicates the *DNS name* or *IP-address* of the image server.

prefix An arbitrary string which can be used to identify a subset, project etc.

identifier An identifier which, together with the prefix, uniquely identifies an image. The IIIF-standard does not impose a special formatting of the image identifier. It may be a number, an archiving signature, a filename, an IRI (url encoded) etc.

region allows to select a rectangular region (e.g. region of interest) of the image, either in coordinates or percentage (upper left corner, width, height).

size allows to select the size (in pixels or percentage) that the selected region should be adjusted to. If both width and height are given, the image will be distorted accordingly. If only one of the dimensions is given, the other dimension will be determined so that the aspect ration will be preserved.

rotation This parameter allows to define an arbitrary rotation and/or mirroring of the selected part of the image.

DOI: 10.2352/issn.2168-3204.2016.1.0.95

quality allows to select B/W, grey and color rendering.

format The file format to be used. Currently the standard includes JPEG (.jpg), TIFF (.tif), PNG (.png), GIF (.gif), JPEG2000 (.jp2), PDF (.pdf) and WEBP (.webp). At least theJPEG and TIFF format have to be supported by a IIIF compliant server.

For both the *region* and *size* the keyword "full" indicates to transmit the full image resp. use the maximal resolution.

Since the syntax of the IIIF request The IIIF-standard is very flexible, many variants may result in the same digital image returned. Therefore IIIF defines a canonical request URL for each request which is added to the response as LINK-header. Thus a client is provided with a unambiguous – thus canonical request URL which will always return exactly the same image.

There are also many web-based viewers available that are capable of displaying high-resolution images using tiling that is supported by the IIIF standard (e.g. OpenSeadragon[2], mirador[3] etc.). Given the dynamic selection of region of interests and resolution, a IIIF server is capable to provide tiles for this kind of viewers.

The IIIF standard does not specify any details of the implementation of the IIIF functionality. However, virtually all IIIF-compliant image servers share some common ground:

1. On the server, only one image file at highest resolution and quality is stored.
2. When accessing an image, the server performs the necessary transformations to convert the image to the required format, region of interest, resolution etc. on the fly.
3. After conversion, the image data is being transmitted to the client using a HTTP response.

Some features which are important in real world application (e.g. for GLAM[4] institutions) are not (yet) part of the IIIF-standard. Most prominently there is no standard defined on the implementation of authentication and access right management. In addition, the IIIF standard does not mention the treatment of embedded metadata such as EXIF-, IPTC- and XMP-headers as well as ICC color profiles.

In addition to provide access to the image data itself, IIIF offers also a standardised way to query image information (metadata). The syntax therefore is as follows:

```
http(s)://{server}{/prefix}/{identifier}/info.json
```

The information is returned as JSON or JSON-LD formatted string and includes width, height, preferred sizes, optimal sizes for tiling etc. In addition the IIIF standard includes a (hierarchical) data model on how to embed digital images into a digital representation of real world objects. This information is also presented using JSON-LD.

Simple Image Presentation Interface (SIPI)

There are several image servers available that either natively support the IIIF V2.0 standard or offer some translation tools[5]. However, most of these image servers do have serious drawbacks and limits. Therefore we decided to implement a high performance, versatile IIIF-image server with a small footprint in C++ which includes some features that will facilitate the longterm preservation of digital images.

- The master files usually consist of lossless compressed J2K images (but other formats such as TIFF would be supported but not recommended). The SIPI server is itself able to convert common image formats (TIFF, PNG or even JPEG) into a suitable J2K image (either on upload or using a command line tool for mass conversions).
- It is crucial that image metadata embedded in the original master image files is preserved both on import and export, including ICC color profiles, IPTC, EXIF and XMP metadata. This goal is achieved by using several open source packages, especially littleCMS[6] for processing of ICC profiles, Exiv2[7] for processing of IPTC, EXIV and XMP metadata.
- The server supports a simple, flexible and efficient disk layout of the master repository. The structure is suitable for long term archival needs.
- Easy integration of existing databases using the embedded Lua scripting.
- Support of additional features such as access permission control, dynamic application of watermarks etc.
- Extension to provide a full RESTful API including multipart-upload for directly uploading and converting images within SIPI Configurable caching, based on the canonical URL of IIIF-compliant requests, to improve performance, especially when using viewers based on tiling.

Implementation notes

In order to achieve the high performance required we decided to do a multithreaded implementation of SIPI using C++11. Currently the JPEG2000 implementation is based on the commercial kakadu-library[8]. Because the IIIF-standard implies that images are being converted on the fly, performance is an important issue. Open source JPEG2000 implementations such as JasPer[9] or OpenJPEG2000[10] are not powerful enough performance wise. The kakadu-library has been extremely reliable and very performant. All other libraries used are provided with open source licenses.

One of the major obstacles is the treatment of metadata within the different file formats. We have concentrated our efforts on three major formats. (1) the TIFF format which is still the most common format for the interchange of high quality images, (2) the JPEG format which is most widely used for dissemination (web etc.) and the JPEG2000 format as internal storage format and for archival purposes.

TIFF The TIFF format will be the major source for files that have to be integrated into the SIPI/IIIF framework. TIFF is an well established, "old" file format which offers a lot of

[2] `https://openseadragon.github.io`.
[3] `http://iiif.github.io/mirador/`.
[4] *G*aleries, *L*ibraries, *A*rchives and *M*useums.
[5] For a list see `http://iiif.io/apps-demos/`.
[6] `http://www.littlecms.com`.
[7] `http://www.exiv2.org`.
[8] `http://kakadusoftware.com`.
[9] See `https://www.ece.uvic.ca/~frodo/jasper/`.
[10] `http://www.openjpeg.org`.

DOI: 10.2352/issn.2168-3204.2016.1.0.95

possibilities and variants. Its structure is extremely flexible. There are two major ways how metadata is being integrated into TIFF:

- Using the TIFF structure of tags. The TIFF itself incorporates some metadata tags such as tag 0x8298 for copyright information or 0x013B for the artist.
- Using a separate Image File Directory (IFD) as is being used to store EXIF data. The structure and layout of the IFD itself follows the TIFF standard.
- Using a binary blob. This is the case for ICC, XMP etc. data which define there one format. In these cases there is a special tag (e.g. 0x8773 for the ICC color profile) which is used to store the binary data.

JPEG The JPEG format stores metadata in file segments delimited by 16 bit markers. The metadata is then stored as blob. The use of the segments has not been formally defined, but there are some common practices. In the implementation of SIPI we did follow the de facto standard use of JPEG markers as provided by Adobe Photoshop. This required some in-depth analysis of files created with Photoshop since the documentation is very scarce and sometimes contradicting.

JPEG2000 The JPEG2000 format includes some provisions to include metadata using special sections (called "boxes") which are identified through uuid's. There is a limited integrated support for ICC color profiles.

When importing an image into SIPI, all the available metadata is extracted and represented in appropriate in-memory structures (e.g. ¡std::maps¿) or in structures given by the third-party libraries (e.g. littleCMS profile structure for ICC colours profiles, exiv2-structures for Exif or XMP. If necessary (e.g. reading exif form a TIFF, the proper structures are being created). When writing a file, the metadata is again converted to the appropriate representation for the given format and integrated into the file stream. Usually, the image representation is not changed on ingest as long as JPEG2000 is able to represent the specific image representation. E.g., CMYK encoded images are stored as CMYK encoded JPEG2000 internally. All necessary conversions are done in the moment a IIIF request is received. For example, SIPI can be configured to convert all images served as JPEG to RGB and the sRGB color profile (which is usually what the browsers are able to handle).

In addition, using the scripting capabilities, it is possible to add on-the-fly metadata form an external database. The overall architecture of the SIPI server is shown in figure 1.

For configuration and integration into existing image server platforms we integrated the *Lua* scripting language[11]. Lua is a modern, small footprint scripting language that has gained a large user community (e.g. in the world of computer game developers) for its efficiency and performance . For each request to SIPI there are pre-, in- and post-flight hooks to Lua scripts. Thus, access permission check, naming schemes, conversions etc. can be processed using flexible Lua scripts.

The SIPI-server implements a very efficient caching. The IIIF-standard defines for all requests a canonical URL, which is stated in the documentation as follows[12]:

> There are many ways in which the same image can be described using the different forms of the parameters. While it is useful for clients to be able to express their requests in a convenient form, there are several reasons why a canonical URI syntax is desirable:
>
> - It enables static, file-system based implementations, which will have only a single URI at which the content is available.
> - Caching becomes significantly more efficient, both client and server side, when the URIs used are the same between systems and sessions.
> - Response times can be improved by avoiding redirects from a requested non-canonical URI syntax to the canonical syntax by using the canonical form directly.
>
> In order to support the above requirements, clients should construct the image request URIs using the following canonical parameter values where possible. Image servers may redirect the client to the canonical URI from a non-canonical equivalent.

The SIPI server creates a user configurable cache (cache size, purge parameters etc.) using the canonical IIIF-URL as index. Before any conversion is done, the server builds the canonical URL from the given request and checks the cache. Thus in many cases the conversion process can be avoided which speeds up the response time considerably.

In addition SIPI contains a web server with a very small footprint which is able to serve static HTML, CSS and JavaScript files as well as HTML with embedded Lua. Thus, without additional infrastructure and minimal effort, admin interfaces, access pages or ingest pages (upload) can created.

SIPI and JPEG2000

The recommendation within SIPI to use the JPEG2000 format for the master file for a IIIF compliant server has several advantages which important for an efficient IIIF implementation:

1. The creation of reduced resolution images if very efficient because J2K allows the usage of resolution pyramids. That is in order to generate a low-resolution version of an image; only a fraction of the master file has to be read.

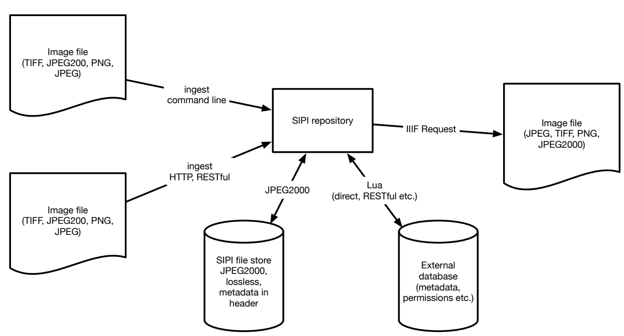

Figure 1. *Architecture of a SIPI as IIIF-compatible image server. Access to an external database for checking access permissions, additional metadata etc. is optional and is being controlled by lua scripts.*

[11]`http://www.lua.org`.
[12]`http://iiif.io/api/image/2.0/#canonical-uri-syntax`.

DOI: 10.2352/issn.2168-3204.2016.1.0.95

2. The generation of tiles (for tiled viewers) is also very efficient due to the internal organisation of the J2K files.
3. The lossless compression allows generating high-quality master images (e.g. 16 Bit RGB-TIFF images) if necessary. One of the major obstacles during the implementation is the lack of standard of storing the metadata across the different file formats. It required a lot of reverse engineering and try and error to find a robust implementation that preserves the essential metadata across all supported file formats. This is also one of the major drawbacks of most existing IIIF servers: metadata embedded in the file header is most often just discarded.

Long-term access and preservation of digital images

Currently, the canon for long-term preservation is to use a OAIS-based archive. However, as I will show below, the OAIS-model is not necessarily and in all cases the optimal archiving strategy.

The OAIS approach to long-term archiving

The OAIS reference model of a digital archive is based on the migration model. In addition to a formal process description, it also covers the ingest of data into the archive and the dissemination of archived data to a user. The OAIS model has been developed by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) which is part of the NASA. It is document oriented where a document may be a single item or a coherent (regarding the content or topic) collection of items such as text documents, data sets, images etc. . An important aspect of the OAIS reference model is the systematic approach to metadata which is distinguished between the metadata necessary to identify and find a document, and the technical metadata necessary for the management of the migration processes. Archives usually deal with data or entire collections (consisting of a series of thematically related documents/images) which are preserved as entities. These entities, enriched by standardised descriptive metadata and additional context data and documents are submitted by the producer to the archive in the form of the Submission Information Package (SIP). There the SIP is transformed and supplemented by technical and administrative metadata to form the Archival Information Package (AIP) which is archived. Finally, if a consumer requests the information, a Dissemination Information Package (DIP) is formed with a copy of selected data and transferred to the consumer. Again, the original data remains unchanged.

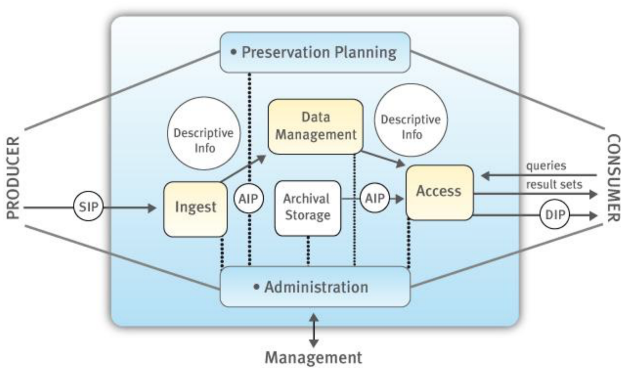

Figure 2. *Schema of the OAIS reference model.*

The preserved data undergo the cycle of periodic migration necessary to overcome technological obsolescence, thus ensuring the continuous use of data and minimising the risk of losing access

It is obvious that such an archive is well suited to preserve digital images for a long time. However, there are several drawbacks, especially for small and medium institutions which do not have large IT departments and/or dedicated digital archiving specialists. One of the major drawbacks is as follows:

An OAIS-archive imposes a dichotomy between *archiving* and *use*. In order to use an archived item (e.g. a digital images), it has to be identified (using the metadata) and a request to the archive has to issued. The archive then retrieves the AIP, converts it to a DIP which is then transferred to the user. This process, even if totally automated, is complex and quite time consuming. Thus, in parallel es second infrastructure has to be built for the easy use of the digital assets. Also this infrastructure has to maintained and backed up. Usually in order to guarantee a high quality of service, quite elaborate backup concepts are being applied.

This doubling of infrastructure is often beyond the economical and technical capabilities of small and medium institutions.

The keep-alive archive and IIIF

Complementary to the OAIS archival process model, *keep-alive archiving* keeps a system of data, data management and access methods online and permanently up-to-date. That is, at any time the technology evolves (e.g. a new stable version of the data management software or a new version of a file format is released), the whole system is migrated to conform to the new environment.

In case of digital images, another problem arises. While most online access to images is based on the JPEG format (that's what all browsers understand), the lossy nature of the JPEG compression scheme makes the JPEG format unsuitable for longterm archival. Most memory institutions therefore rely on the TIFF format which is well proven for long term archival of digital images (e.g. [2], [1]). However, since the TIFF format is not well suited for dissemination, more complexity arises.

The JPEG2000 format is in certain respect an ideal candidate for storing digital images. Most important there is the possibility to use a *lossless* compression mode. The JPE2000 format has been proposed by several authors for long term archival (e.g. [3], [4] etc.). One of the disadvantages is that the JPEG2000 format is very complex (compared to the TIFF). However it is interesting to note that the JPEG2000 format has been since considerable time been accepted as standard format for the movie industry ([5]). The Digital Cinema Pack (DCP) [13] [14] which is todays standard of distribution of movies to the theatres is based on the JPEG2000 format. Therefore it is probably not wrong to state that today largest number of high quality digital images are stored in the JPEG2000 format. That is, despite being a complex format, JPEG2000 is in widespread use and will not disappear on a short scale. There are many products available which offer JPEG2000 implementation, even some common browser (e.g. Apple Safari) offer native support of the format.

[13] `https://en.wikipedia.org/wiki/Digital_Cinema_Package`.
[14] `http://www.dcpinfo.com`.

DOI: 10.2352/issn.2168-3204.2016.1.0.95

Thus, an image repository using SIPI (or another IIIF compatible server which preserves metadata) is able avoid the need to duplication of an archival and access infrastructure. The same infrastructure can serve both for archival and access at the same time. The versatility of SIPI make it an ideal candidate for such purposes. In adhering to the IIIF standard for access as well as using the lossless variant of JPEG2000, both the needs of access and long term archival can be satisfied without duplication of resources. This should facilitate the burden of both tasks for small and medium institutions.

Applications of SIPI

We are currently using SIPI in the context of Knora ("Knowledge Organization, Representation and Annotation")[15] and SALSAH (System for Annotation and Linkage of Sources in Arts and Humanities)[16]. Knora/SALSAH form together the IT-framework of a national center for the long-term preservation of digital research data in the humanities ("Data and Service Center for Humanities") that is being built by the Swiss Academy of Humanities and Social Sciences (SAHSS). Knora itself is a repository based on semantic web technologies (RDF, RDFS and OWL) and offers a RESTful interface for easy access. SALSAH is a generic web-based front-end implementing a virtual research environment for the humanities. Since today most of the research projects in Humanities rely to a large part on digital objects as primary sources, digital images of artefacts, manuscripts etc. need to be preserved together with the annotations, links etc. Therefore the use of the IIIF compliant SIPI server allows us to facilitate the long-term archiving to a great extent:

- the lossless compression reduces the weight of the master files by a typical factor of 2.3.
- Simple copies of the SIPI repository can be used for long-term archival of the images (e.g. using rsync to pull copies from archival hard disks to LTFS magnetic tapes). Apart from the metadata that is included in the file header, additional metadata could be provided as sidecar (e.g. XMP) files if necessary.

Conclusion

SIPI provides an IIIF compliant image server which is ideally suited for a context where the long-term preservation of image data is necessary. There is no longer a need to use different formats for presentation and long-term archival since SIPI can deal directly with the archival format as the image server implementation is capable to handle important archival file formats as well as metadata standards for technical and contextual meta-information in an environment with high need regarding performance, permanence and reliability. SIPI will be put into open source by early sprint 2016 and will be found on github.

Acknowledgments

We grateful acknowledge the continuous support from the Swiss Academy of Humanities and Social Sciences (SAHSS) and the Swiss State Secretariat for Education, Research and Innovation (SERI) which made possible to develop SIPI and offer it as open source.

[15] `http://knora.org`.

[16] `http://salsah.org`.

References

[1] R. Gschwind, L. Rosenthaler, R. Schnider, F. Frey, and J. Frey, Digital Image Picture Archives: Theory and Practice in Switzerland, in Digital Applications for Cultural and Heritage Institutions, pp. 123 132. Ashgate Publishing Limited, 2005.

[2] Lukas Rosenthaler, Digital Archiving, in: Michael R. Peres et al., FOCAL Encyclopdia of Photography, 4. Ed., 359 364, Focal Press/Elsevier, 2007

[3] Robert Buckley and Simon Tanner, JPEG 2000 as a Preservation and Access format for the Wellcome Trust Digital Library (2009). `http://library.wellcome.ac.uk/assets/wtx056572.pdf`

[4] P. Desrochers, and B. Thurgood, JPEG 2000 Implementation at Library and Archhives Canada, in J. Trant and D. Bearmans (eds.) Museums and the Web 2007: Proceedings, Toronto: Archives & Museum Informatices, published March 1, 2007

[5] M. W. Marcellin and A. Bilgin, JPEG 2000 For Digital Cinema, SMPTE Motion Imaging Journal, Vol. 114, No. 5-6, pp. 202-209, May/June 2005.

Author Biography

Lukas Rosenthaler is the head of the Digital Humanities Lab of the University of Basel. He also leads the Swiss national data center for providing long-term access to digital research data in the humanities, an initiative of the Swiss Academy of Humanities and Social Sciences. He is an expert for image analysis, image processing, data base systems, linked open data, virtual research environments and digital archiving and he is supporting open access initiatives.

Peter Fornaro is deputy head of the Digital Humanities Lab of the University of Basel. He is doing research in the field of digital archiving, imaging, cultural heritage preservation and computational photography. Fornaro is teaching at the University of Basel. Besides research and lecturing he is giving consulting to companies, archives and museums