Specification Document

Virtual Light Table

Crossing Boundaries
V.0.1 (SP & SU)

The Virtual Light Table (VLT) is envisioned as a digital tool for scholars to browse, select, load, compare and join various papyrus fragments (available as 'objects' in the TPOP), the ultimate goal being to reconstruct virtually entire documents. The basic requirements for this tool are listed below. The ones listed in **black** are top priority, the one listed in **grey** are for a second stage of implementation, and the ones listed in **blue** will be developed by members of our project. A tentative drawing of the interface as we envision it is provided at the end of the document.

1. **Access to objects**.
   Users need a method to browse the available papyrus fragments stored in the database, by scrolling through a list of available objects visualized as snapshots. On mouse over, the image should be enlarged and information about the fragment should be displayed: inventory number(s) and text_content>type(s) + hyperlink giving access to the fragment with all its metadata in TPOP (other tab of the browser).

2. **Filtering available objects**.
   Users need to be able to filter the list of fragments using one of two methods: (a) manual filtering or (b) automatic filtering. The second option will be developed by Stephan Unter and is therefore outside the scope of the present specification.

   a. **Manual filtering**.
      If users select 'manual filtering', they should be able to filter the list of objects according to **(i) features of the object**:
      i. inventory number(s),
      ii. features>colour,
      iii. sheet join>yes/no,
      iv. folds>yes/no,
      v. palimpsest>yes/no,
      vi. status,
      vii. and general description[full text])

      or **(ii) features of the witness(es) on the object**.
      viii. features of writing
         1. Colour of ink,
         2. Punctuation (yes/no)
         3. Tables/lines (yes/no)
         4. Checkmarks (yes/no)
      ix. Scribal practices and layout

1. Size of hand
2. Style of hand
3. Interline

x. Text content
1. Text type
2. Keywords
3. Personal names
4. Deities

xi. Drawing
1. Drawing (yes/no)
2. Colour
3. Grid (yes/no)

xii. Date
1. Dynasty
2. Pharaoh

b. **Automatic filtering**. If the user select this option, he should be presented with objects that, according to the machine learning algorithms, match the fragments already loaded in the VLT (see next point). The fragments should be ranked according to probability of connection. If possible, the interface should also suggest joints, including fitting visualisations of results

c. In the list of fragments that match the filtering criteria, the fragments that are already loaded in the current document should be framed in green (one cannot load a fragment more than once in one document: this would indicate that they are already loaded), and the fragments that belong to other objects should be greyed out (in order to show that they are not the most likely candidate for a new document.

3. **Load fragments**.
From the list of objects, users need to be able to load selected fragments into the light table environment (on right-click > load object in VLT). The **background of the loaded image** should be **deleted** automatically in order to allow for a precise positioning of the fragments in the VLT (could potentially be done by using machine learning algorithms).
   a. The object loaded in the VLT should not overlap with any other object already loaded in the VLT: it should be positioned automatically next to other fragments. This means that the size of the VLT has to adapt to every new loaded fragment.
   b. For each fragment, one should be presented with the choice to add it to the current layer or to a new layer of the VLT (see below under 4.c).

4. **Organizing the fragments**.
Once a fragment is loaded in the VLT, the following capabilities should be implemented:
   a. **The scale of the fragments** should always be proportional to their actual size. Users should not be allowed to modify the scale of individual fragments.
   b. **Playing with individual fragments**

i. **Moving fragments**: users should be able to position the fragment in the VLT where they want.
ii. **Rotating fragments**: 90°, 180°, or freely. Users need to rotate fragments independently from other fragments
iii. **Flipping fragments**: r° > v° and vice versa. Users need to be able to flip single fragments such that the outline of the fragment is mirrored and the opposite side is rendered.

c. **Creating layers/groups of fragments**. Much like in Photoshop or Illustrator, it should be possible to group together fragments together in layers, to freeze individual layers, and to hide/visualize layers independently.

5. **Functionalities of the VLT as a whole**
   a. **Save/load documents**. Users need to be able to save and load documents. When saving a document, all the loaded fragments must be saved plus their current (relative) position, rotation, orientation, etc. The saves should be stored at server-side, thus allowing for other users to browse the existing documents and for everybody to operate on the scenes from different locations/devices.
   b. **Document Save History**. Whenever a document is saved, older versions of this particular document should be kept such that users can go back to older versions or track changes over time.
   c. **Zoom**. Users need to be able to zoom in and out; all the fragments need to be zoomed in/out and the position of fragments relative to each other must remain unchanged. Users should be presented with information about the current scaling of the scenery, and a mini-map (top right-hand corner?) should display the position of the zoom in the entire VLT.
   d. **Scene and objects overview**. Users should have the option, e.g. via a specific button, to readjust the viewport of the VTL such that all fragments are visible at the same time. In other words, with a click on that button the viewport of the VTL would represent the minimal bounding box of all loaded fragments. Users should also be able to select in a drop-down list one of the object belonging to the document and, thereby, to have the zoom of the VLT positioned on this specific object.
   e. **Hide/show menus.** It should be possible to hide all the menus (left-hand side, top, bottom) in order to have the full screen available for piecing fragments together and visualizing the document.
   f. **Table Flip**. Users need to "flip" the table such that all fragments are flipped at the same time; fragments need to stay in relative position to each other.
   g. **Collaborative interface**. Multiple people could view the same document and interact with fragments belonging to a single document.
   h. **Export function**. Users need to able to export (a) the current viewport or (b) the whole scene as an image file, e.g. as .jpg, .tif, .png file. Potentially additional information to the visualised fragments, at least a list of their IDs should be easily exported, too.

Kommentar [1]: "snapshots" - could be linked later on by TPOP such that visitors can look at the reconstructed documents

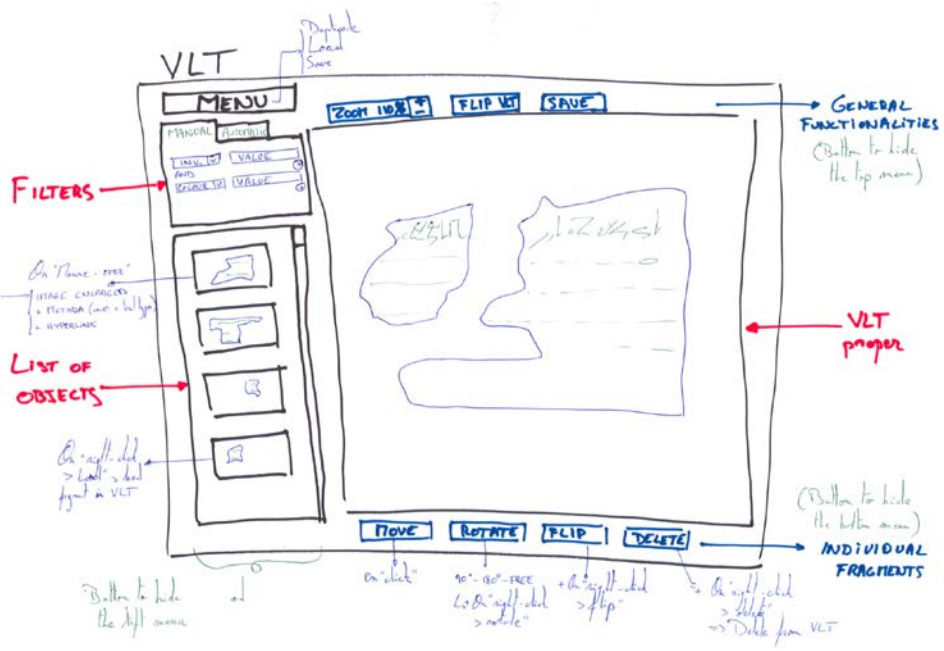
Kommentar [2]: Maybe also export into TPOP? Would mean at least transfer of joining fragment IDs into TPOP, including an access check.

6. **Design of the platform**
    a. The final tool has to be made available as an **open source solution**. Although it will be directly plugged onto TPOP, it should be made available as an independent solution.
    b. It is envisioned as **a web-application**, with **API access**. It must be designed in such a way that it is easy to get information about selected papyri from the table or to enter probability results to the table. Potential connectors: scripts for providing database access (especially in open source case with databases other than TPOP), machine learning algorithms, user interfaces...
    c. The design of the VLT is thought of as a **server/client architecture**: while the server is establishing and maintaining the connection to the database, gathering resources and processing them, the client is responsible for the visual user interface presenting the data and providing interaction capabilities. For Crossing Boundaries, the server (NodeJS?) could be running on the museum's servers or on a virtual machine (server) of one of the universities involved in the project. With respect to the open source approach, the server could be run by anybody with respective capabilities, too. This also allows to add further user interfaces if needed at a later stage, e.g. for the usage of the VLT as a mobile application.
    d. The **web technologies** used for the VLT should be as **open, flexible and commonly distributed** as possible, especially as the software has to be released as an open source product at the end. (mentioning IIIF as recently developed open framework for more sophisticated image representation)

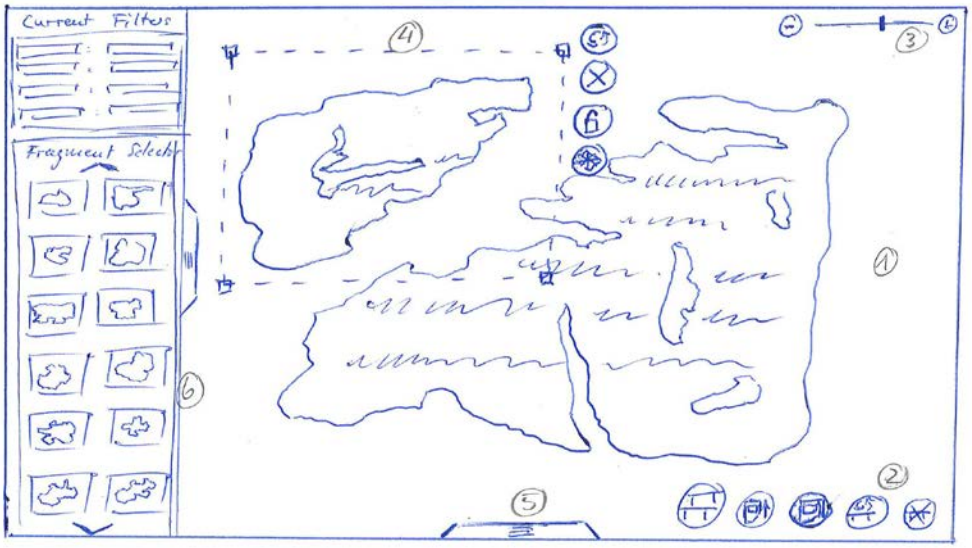
Features which might be interesting for further development (brainstorming):

- If multispectral analysis proves to be useful: modes to view different lighting settings?
- Overview/"minimap" (esp. for large documents?)
- Witnesses and their relation to Object/Documents

Basic functionalities of the interface



Extended version



(1) Virtual Light Table. Ideally, the whole viewport of the device is used for the light table, thus granting the user as much research space as possible.

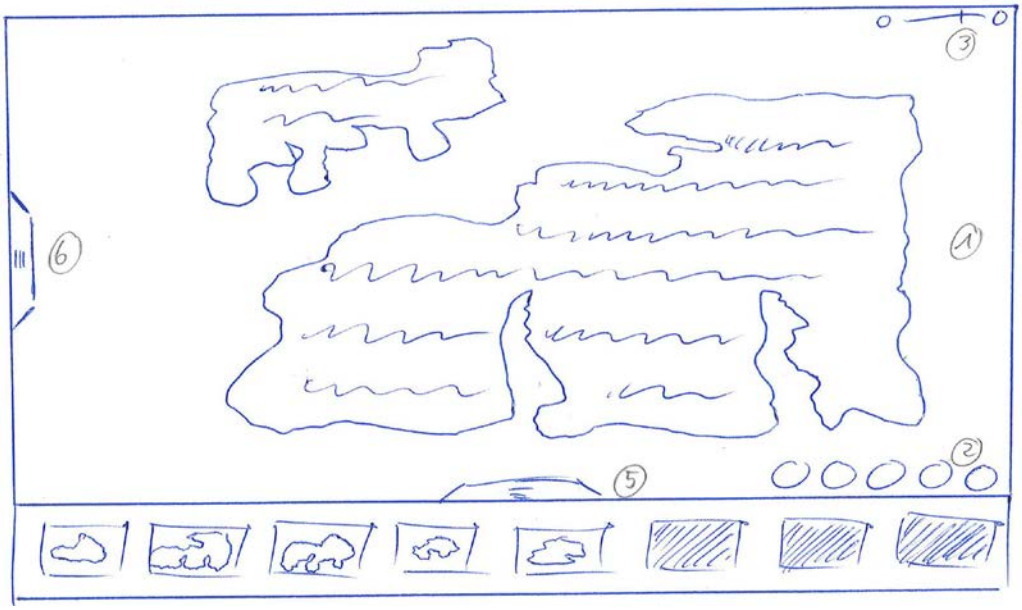
(2) **General Functions**. These functions affect the whole table, i.e. all fragments loaded into the scene. From left to right (note: the symbols are NOT final):
   (a) Duplicate table
   (b) Save scene
   (c) Load scene
   (d) Flip Table
   (e) Clear Table

(3) **Zoom Function**. The user can zoom in and out either stepwise (by using the buttons) or more flexibly by using the slider.

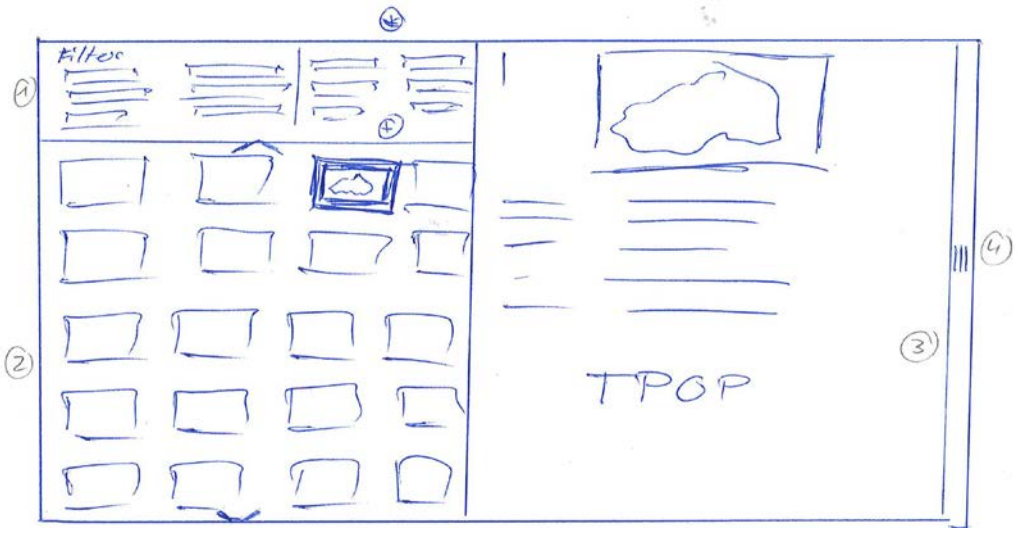
(4) **Selected Fragment**. A fragment can be selected to further manipulate it. The selection must somehow be indicated visually, either by a surrounding box or maybe by outlining the silhouette in a bright color. When selected, additional (floating) buttons appear, giving further options for manipulation:
   (a) Flip fragment
   (b) Remove fragment
   (c) Lock Fragment
   (d) Move Fragment
   (e) [Rotate Fragment] (necessary if the selection is only indicated by outlining)

(5) (hidden) **Tray of Loaded Objects** (detailed see picture 2)

(6) **Fragment Selector**. This left-hand side menu bar contains the fragments loadable from the TPOP database and an area to filter the viewed results and thus improve the results. Results from the machine learning process have to be included, too - probably by probability values, sorting, color coding? In the top, the current filters are viewed (and can be changed). In the bottom, fragments fulfilling the filter criteria are viewed. Fragments can be loaded into the scene by either Drag&Drop or a special button. The Fragment Selector can be extended by dragging the handle to the right (see picture 3).



(1) Virtual Light Table.
(2) General Functions.
(3) Zoom Function.
(4) (not visible here)
(5) **Tray of Loaded Objects.** Here all fragments already loaded into the scene are displayed. The user can either gather more information about those fragments (see picture 4) or directly move the scene towards a specific fragment by clicking on its representation in the tray. The handle either opens or closes the tray.
(6) (hidden) Fragment Selector.



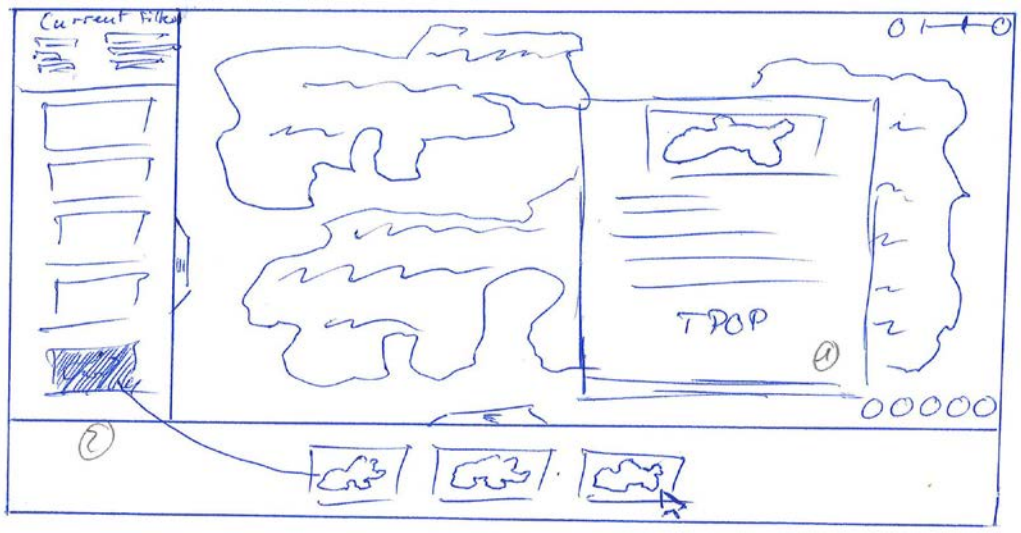
(*) This area is not yet clear and needs further work.

(1) Extended Filter Section. This section is not yet clear, speaking of design/arrangement of elements.

(2) Extended Fragment Selector. Purpose: Showing more fragments at the same time.

(3) Detail View for Individual Fragments. Here all information about an individual fragment is shown as given by the TPOP database (or whatever database is behind the research conducted).

(4) Close Handle. By using this handle the extended view can be closed again, leading the user back to the original VTL view.



(1) **Detail Overview**. When the user hovers the cursor over a fragment's thumbnail, a small overlay will be shown (floating above the Virtual Light Table) showing more detailed information about the fragment derived from the TPOP database. Due to the size of this window, the information is probably not as complete as in picture 3, maybe (hopefully?) there are bits of information which could be removed.

(2) **Greyed out fragments**. Fragments which have already been loaded into the current scene need to be framed in green in the fragment selector. They should not be completely removed, as they still fulfill the filter requirements, but they cannot be loaded into the scene twice/multiple times.