

Decapod Workflow



This workflow needs to be updated with respect to the v1.2 User Interaction Workflow. -JH. Feb 3, 2010.

Introduction

This document describes the user interaction with the Decapod system, and technical back-end activity that occurs in Decapod. Structuring this information as a start-to-finish workflow is a way of portraying Decapod helps serve as a work plan for both design and development, as well as provide a conceptual view into how Decapod functions as a whole.

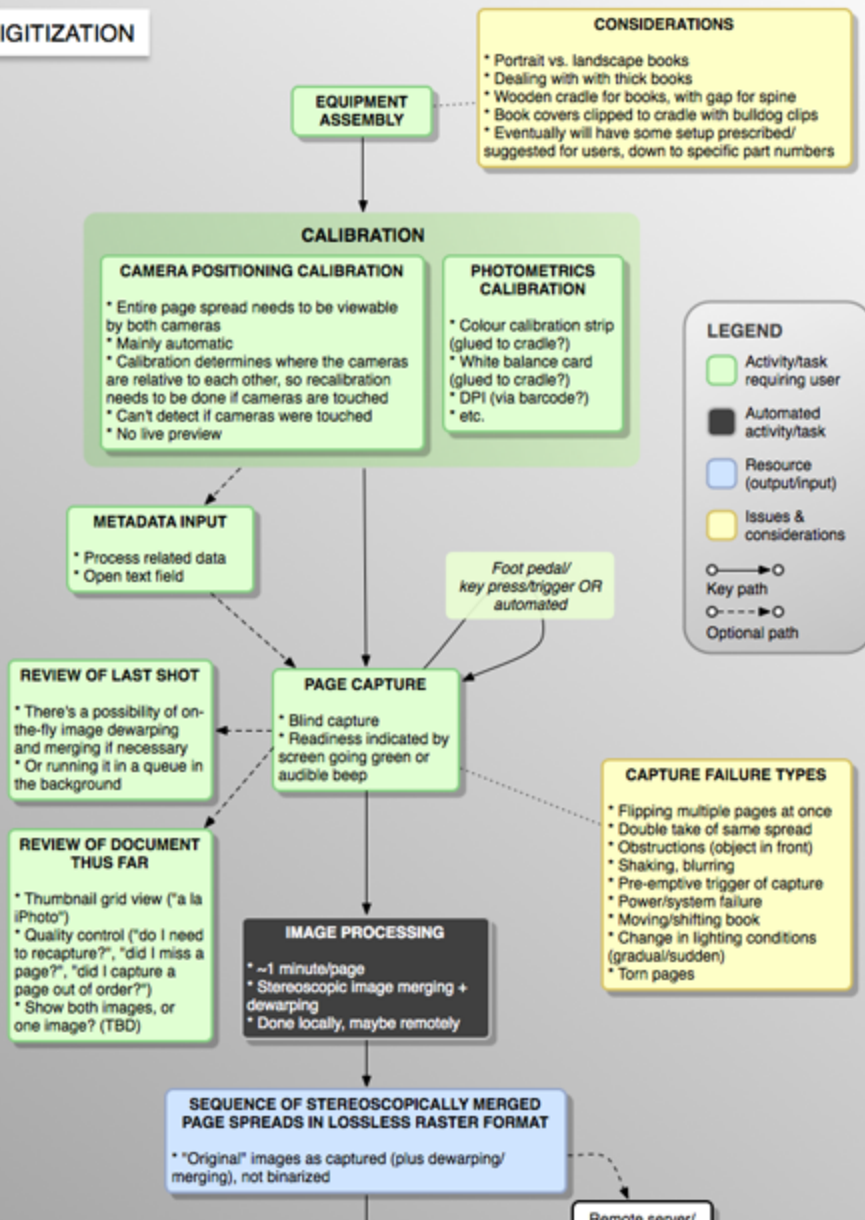
If you are new to Decapod, this document may be heavy on details, but gives a good description of how the system will work, and where we are going in our work.

Digitization Process Diagram (Draft 2/August 6, 2009)

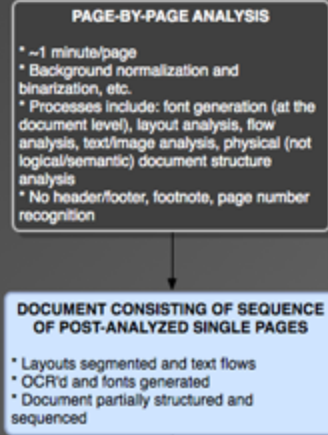
The following is a graphic illustrating the high-level workflow. More detailed descriptions follow further in this document.

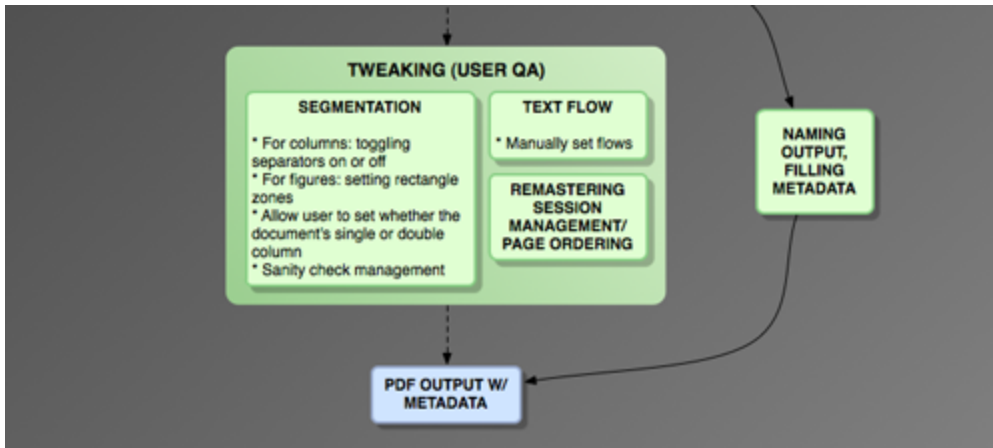
Decapod Book Digitization Process

DIGITIZATION



REMASTERING





[Download PDF of digitization process diagram, Draft 2 \(August 6, 2009\)](#)

[Download PDF of digitization process diagram, Draft 1](#)

Overview of Workflow for User and System

1. Start Decapod
2. Assemble hardware if not a fixed installation (Wireframe: [Camera Setup Wizard](#))
3. Calibrate (Wireframe: [Camera Setup Wizard](#))
4. User begins capturing (Wireframe: [Detailed and Thumbnail View](#))
 - Capture to memory card or direct to USB
 - Download to system
 - Pre-process for user presentation:
 - image scale / downscale
 - image crop
 - normalization
 - Present pre-processed results of page spread stereo pair (Wireframe: [Detailed and Thumbnail View](#))
 - In background, Dewarp and page splitting is being applied to unprocessed images
 - Pairs are merged and spreads are split.
5. User manages individual pages for Exporting (Wireframe: Future work)
6. Indicates they want to Export (Wireframe: [Menu Bar](#))
 - Page Segmentation and Document Analysis occurs. Can take a long time.
7. Output generated to PDF. (Wireframe: Future work)

Start Decapod

- Wireframe: [Decapod Dashboard](#)
- Load Decapod (doesn't matter if fixed/mobile system, or if booted by LiveCD or installed)
- If user using Ubuntu Live CD, user should be prompted for a storage location.
- User presented with options to start a new project or continue an old one.
- If starting a new project, user will be prompted for Metadata*
 - who, where, when, general comment field
 - Motivation: Sometimes detailed metadata is not available (old text, obscure indexing, little/no instruction by staff), therefore simple fields that any user can answer is all we're requiring. More detailed metadata can be filled in during Remastering.
- If required, a help option is available to show how to assemble hardware.
- The "Project Manager" UI may appear at this step in the workflow.
- Example of Server/Client application in Ubuntu distro: SageMath - server/client application.

Hardware Assembly

- See [Decapod Wireframes](#)
- This step is only necessary if using a portable Decapod system
- Mark the center using the UI for page splitting.
- Avoid any construction
- Gaylord book library display (<http://www.gaylord.com>)
- A welcome screen to ask user where they would like to store their information.
- Required parts: Decapod system, 2 matching Decapod supported cameras, tripod, accessory bar
- Optional parts: cradle, foot pedal


Calibration

- [See Decapod Wireframes](#)
- gphoto detects cameras are attached. Works well with subversion version of gphoto.
- move a target around the view and outputs an XML file with calibration information
- Then take photos and dewarping applied to image.
- Can detect drifting of pages.
- Need some way to help position cameras and book in Decapod.
- Add colour and grey cards to calibration process
 - Cards can be glued to cradle to simplify process.
- Produce a short user guide into how to set up and mark the book position and cameras:
- Can use transparent tape to mark location of book edges.
- Can also use bull-nose clips to clamp down book (but could be damaging).
- Have the ability to identify cameras and swap them if the left camera is not appearing properly in the left image review.
- there is no need to mark the reference plane or boundaries or use a different coloured mat since the calibration process does not require it.



The process:

- place calibration target viewable by both cameras
- System captures images to be used for calibration
- Once sufficient number of images are taken (i.e. 4 to 8 images), the system will report back to the user and the user can begin their work.
- see [video of Calibration Process](#)

Capturing

- [See Decapod Wireframes](#)
- Future development: Quality Control plugin that does partial detection to help improve responsiveness and UX
- Errors are generally non-descriptive because the nature of the beast has a lot of varied errors. Still needs to be communicated to the user.
- Memory card for cameras are optional. gphoto can capture directly to server storage.
 - it used to be the case that using memory cards as intermediate storage was more reliable to write to card and then transfer to computer.
- gstreamer - image capture library for gnome. Can do realtime capture.
- gphoto - talks to photo cameras. Use SVN version for better camera support and direct capture to usb.
- Real-time camera preview - not deal with preview at the moment because it's only used for positioning. Really need to position once and clamp down.
 - Also preview is not a priority because it's not reliable across all brands and models, and there may not be a high demand for such a feature.
- Blind capture: so users can concentrate on the manual task. Therefore keep capturing UI very simple.
- Prompt for readiness
 - visual and audio prompts. Visual prompt: green/yellow/red (consider accessibility and colour casting). Audio prompt: chimes, beeps, etc.
- Foot pedal or alternative input to activate the camera, thus freeing the hands to manipulate the book.
- Capturing the spine of a book - no technical implications (i.e. focusing), just like any other capture for the book.
 - Will not be explicitly written into the user's workflow because not every institution will want to capture the spine.
-  Drag and drop support?
 - User drags images from their desktop and drops it into the decapod window.
 - This is a way of accomplishing splitting and merging projects together to form a logical collection of pages (i.e. a "book" or "volume")

Capture Post-Processing

- **Wireframe:** [Thumbnail Interaction and Behaviour](#)
- **Wireframe:** [Detailed and Thumbnail View](#)
- Dewarping involves two steps: combine stereo pair and apply the 3d model to dewarp. The output is a single image with two facing pages.
- Dewarping is costly. 15 second to few minutes depending on the number of reference points required.
 - 1 image is picked as the 'primary' (in Decapod's case this will always be the Left camera) and the second image of the pair will be warped / dewarped to match the primary.
 - This is done by overlaying the 2nd image on top of the primary and calculating the shifts.
- Normalization will occur here as well.
- Splitting individual pages from a spread: user will have to open the book to the middle during calibration and mark the center/split. Then after Dewarp, the pages can be split and presented as separate pages to the user.
- If dewarp doesn't work , then recapture
 -  What does it mean "doesn't work" (i.e. the dewarped output just looks bad?)
 -  From a UX perspective, how does recapture fix a problem if initial capture was bad? Capturing can be separated by time, space, and configuration - so a recapture may provide different results.
- Dewarping process runs continuously hidden from the user:
 - visually indicate to the user that some images are not dewarped-unsplitted, and indicate when those pages have been processed.

Quality Control / Image Management

- **Wireframe: Zoom**
- **Wireframe: Thumbnail Cut and Paste**
- **Wireframe: Reordering**
- **Wireframe: Hiding Thumbnails** (to be designed if there is a use case for this)
- Can display downscaled version to screen with zoomed regions to assist in QA
- Can also show the stereo pair with zoomed regions.
- To display a single dewarped page requires that it goes through the Dewarp process.
- Allow for the user to cut and paste between Decapod documents.
- On any given book, the workflow may be divided by space and time (i.e. multiple locations, different people, over a period of time)

Data Storage / File Formats

- **Wireframe: Menu Bar**
- Each image needs to have associated calibration data - so that if it's moved between projects, calibration data goes along with it.
- About 10GB / book
- Camera captures in JPG directly to USB, intermediate working format is PNG, output format is PDF or TIFF.

Close the Session

- Display Optional Metadata input again
- Dealing with files:
 - Shipped to remote server or locally,
 - or Direct to Remastering process locally or remotely
- Use a flash drive or input field to specify where to upload images.

Remastering for Output



Interaction has not been finalized yet.

There is no real user interaction per se, but likely some notifications so users are aware of what the system is doing.

Wireframe: Future work

Input: Sequence of page spreads

- Already dewarped, aligned, normalized (brightness, contrast, white balance), cropped

High-level workflow:

1. page segmentation
2. line segmentation
3. document flow and hierarchy analysis
4. character segmentation - clustering - tokenization

- Slow part of the process is character segmentation and clustering

Output: Appropriately split pages, OCR'ed, segmented, flowed, font generated, and with proper document structure

Page Segmentation

- No real interaction. Likely status messages.
- Errors are generally non-descriptive because the nature of the beast has a lot of varied errors. Still needs to be communicated to the user.
- Page segmentation algorithms are pluggable - user can theoretically choose their algorithm depending on the reliability / simplicity.
 - Algorithms provide trade-offs in speed and accuracy.

The Tokenization Process:

- No real interaction. Likely status messages.
- greyscale images goes into binary converter for PDF generation
- couple hours per book
 - analyzes structure of page
 - analyzes characters of the page
 - group characters together and generate a font
 - separate images from text

- detect text lines.
- minute per page on a mid-range machine

Layout Detection Process:

- Automated process. Corrections to layouts is done after this has run.
- how long does page segmentation take? About 10 sec per page.
- changes the greyscale into an RGB image - R for columns, G for paragraphs, B for line numbers
- Ocropus - pixelwise analysis, pixel based formats, fileformats on Ocropus home page.
 - https://docs.google.com/View?id=dfxcv4vc_92c8xxp7
- Common error: 2 column poage that gets interpreted as a single column. To fix you need to manually correct per page and save it.
 - Graphically you can use distinctive colours to easily identify different layouts in a thumbnail view since each pixel has an RGB value corresponding to columns, paragraphs, and line numbers.
- Page number detection: serves no purpose aside from helping the user order pages for PDF output.

OCR

- Marginalia will appear, but ignored in OCR
- Headers and footers detection - included in output, but can be removed. Removing headers and footers can be complicated and not fool-proof, so best to keep headers/footers intact.

Quality Control

During Capture

Wireframe: Furture work

- Error handling
 - Communication error: If download or communication takes longer than 2 seconds.
- turn two pages - Provide a way for the user to zoom/preview page numbers to discern proper ordering and missing pages. Possible to do programmatically after OCR has been performed and page number recognized, but out of scope.
- ~~double take~~ Possible to do programatically after OCR has been performed and text compared, but out of scope.
- obstructions - Possible to do programmatically, but easier to provide a UI that easily allows user to discern obstructions (i.e. a thumbnail view with a thumbnail size option). Also can do skin detection.
- ~~pre-emptive foot trigger~~ out of scope
- power failure - can be mitigated by periodic saving.
- change in lighting - compensated through normalization step in capture.
- ~~tern pages~~ Hard to detect. Out of scope.
- shaking - provide a way for the user to discern quality (detailed view).
- moving/shifting book - same as shaking: provide a way for the user to discern quality (detailed view).
- Some automatic quality control: focus, contrast, exposure - some of this is already fixed through normalization. Possible because QC assumes text content.

During Remastering

Layout Correction

- Can change reading order and layout structure.
- Images within text: image anchoring is hard to do, but possible. Out of scope for project.
- Assistive Layout Analysis - click / point on things that are wrong and keep clicking until it looks right.
 - But best standard method is drawing boxes

Possible functionality

- Segmentation correction and Flow correction by drawing boxes / reclassifying identified page items.
- Page Sequence Editing
 - Page ordering by drag and drop
 - Delete, replace, insert missing pages
 - Specify numbering of pages

Technically possible, but takes resources. Possibly beyond scope of project.

- Flow correction:
 - Choose from high probability, or add/edit/delete labeling, add/edit/delete regions manually
 - Choose from alternatives or manually draw or label flow (like inDesign reflow).

Output

- **Wireframe:** [Menu Bar](#)

Scripts run:

- build HTML
- build PDF

- Assertion: one project generates one PDF.
- Should be a way to batch export projects. This likely means a separate interface
- Export artifacts: PDF, Greyscale Multipage TIFF
- Provide a way of naming the output (and other metadata?)

Remastering Notes / Discussions

Wireframe: Future work

- Will likely need a progress indicator during analysis (hours per book, status reports)
- Font generation: <http://en.wikipedia.org/wiki/Potrace>
- design considerations: progress indicator and status reporting. Graphical representation of activity that is occurring on actual thumbnails.
- allow user to set threshold for sanity check + correct for Flow and Segmentation.
- Need hooks to manipulate Segmentation and Flow for QA purposes.