

CARDIFF  
UNIVERSITY

PRIFYSGOL  
CAERDYDD

---

IMAGE ANALYSIS FOR MUSEUM  
INSECT DRAWERS

---

BY LOUISE EVANS

---

---

*Module: CM3203 – One Semester Individual Project*

*Supervised by: Paul Rosin*

*Moderated by: Xianfang Sun*

*5th May 2017*

## ABSTRACT

---

The project in this report has been conducted upon request from the entomology department at National Museum Wales, Cardiff (AC-NMW). Entomology is the study of insects, of which the Museum has a large collection of specimens. The insect collection consists of around 700,000 specimens, with an approximately even split of British and non-British.

The aim of this project is to automate the digitisation of their insect collection by extracting text from high-resolution insect drawer photos. The insect specimens are stored in unit trays and slats within approximately 7500 drawers. The drawer contents are organised by species with a number of corresponding labels, such as each insect's scientific name and author name, which could be detected and transformed into XML data format. Extracting these data will require a number of visual computing techniques and may be improved by constraining the problem using the specific tray layout used within AC-NMW.

The availability of this collection to the public could help in a wide variety of applications which require examples of insects which may be found at the Museum. A digital resource would simplify the process of discovering the specimens available by searching for keywords found within the insect tray images. This topic may be applicable on a global scale and could be of interest to a number of other museums exploring similar digitisation processes. The outcomes of this problem, if it can be successfully resolved, could be shared with other museums to encourage online sharing of data between all entomology scholars.

## ACKNOWLEDGMENTS

---

I would like to thank the department of entomology at National Museum Wales, Cardiff for the resources provided to help with this project. I especially thank James Turner, as without his time and knowledge this project would not have been possible.

I would also like to thank my supervisor Paul Rosin for his support and guidance throughout the project development.

And finally, I give thanks to my family and friends for their help and encouragement when I needed them.

## CONTENTS

---

1. Introduction	1
2. Background	2
2.1. Similar Entomology Projects	2
2.2. Other Related Topics	3
2.3. Relevant Algorithms and Applications	3
3. Specification and Design	6
3.1. Program Requirements	6
3.2. Program Structure	7
4. Implementation	9
4.1. Column Segmentation	9
4.2. Selecting Regions of Interest	9
4.3. Specimen Segmentation	10
4.4. Character Recognition	13
4.5. String Output	14
4.6. Dictionary Matching	14
4.7. Metadata Extraction and XML Output	15
5. Results and Evaluation	17
5.1. Hough transform	17
5.2. Template Matching	18
5.3. Dictionary Matching	19
5.4. Specimen Detection	19
5.5. Additional Evaluation	20
6. Future work	21
7. Conclusion	22
8. Reflection on Learning	23

## FIGURES AND TABLES

---

*Figure 1: Example specimen and label layout of an insect drawer*

*Figure 2: Flow chart displaying program design*

*Figure 3: Hough transform code used for line detection*

*Figure 4: Identification of weak edge (light blue) and strong edge (dark blue) regions*

*Figure 5: Specimen segmentation - basic thresholding*

*Figure 6: Edge pixel (1) and internal pixel (2) illustration*

*Figure 7: Specimen segmentation - removal of linear components using polygonal approximation*

*Figure 8: Specimen segmentation - before and after intersection with detected edges*

*Figure 9: Specimen selection - sparse region removal*

*Figure 10: Segmenting detected text regions using empty rows (1) or decrease in row width (2)*

*Figure 11: Example JSON insect dictionary structure*

*Figure 12: Sobel edge detection output, before and after thresholding by percentage of vertically aligned edges*

*Figure 13: Oversegmentation vs accurate segmentation of column boundaries*

*Table 1: Program testing high-resolution image set*

*Table 2: Sobel and Hough threshold value analysis for images with seven column boundaries*

*Table 3: Levenshtein edit distance results for all insect trays (5918 characters)*

*Table 4: Specimen segmentation improvements applied to example image with 298 specimens*

*Table 5: Accuracy based results of specimen region extraction*

## 1. INTRODUCTION

---

With an increasing interest in open access to scientific data, museums worldwide have been developing methods of sharing their natural history collections using the internet. This requires digitisation of their collections in order to make them available online, providing the general public with free, unlimited accessibility. This has been a particularly important advancement within entomology, the study of insects, where specimens are particularly fragile and require delicate handling, as it provides a closer look at species without risking damaging them. Current advancements in digitisation range from simple imaging of full insect trays (Mantle, LaSalle and Fisher, 2012) down to interactive 3D imaging of individual specimens (Olsen, 2015).

National Museum Wales, Cardiff (AC-NMW) is working towards creating an online entomology collection as a method of sharing their specimens with the general public. The idea for their collection involves creating a web-based inventory using high-resolution images of insect trays from their repository. While the images alone are extremely useful, finding a specific insect within the thousands of insect trays could be challenging and time consuming without a way of searching for a specific insect. But this could be solved with the aid of associated data for each image.

The species within each tray could be used alongside the images, providing a means of searching by specific keywords and greatly improving the functionality of their online database. However, the ability to produce these data would require compiling the contents of approximately 7500 insect drawers into text by copying out each of the labels within them. This process would be both slow and at risk of human error if conducted manually, but the text extraction process could be simplified with the use of computer analysis in order to automate the process.

The purpose of this project, therefore, is to find the means to automate the process of digitisation of museum insect drawer images using digital analysis in order to extract data from within the insect drawer images. The aim is to create a program which can analyse insect trays, including the layout and text labels, to output the detected text with optimal accuracy. The final program should be able to optimise the text extraction using visual computing techniques, outputting detected text into XML data format, which can be used alongside the insect tray images.

The components required to complete this program include high level structural analysis, character detection, dictionary comparison, conversion into XML, among other tasks.

This project will implement techniques studied at university alongside additional research topics, applying relevant ideas and suitable understanding of visual computing algorithms. The use of logic and problem solving skills should help to tackle issues that may arise over the course of the project.

## 2. BACKGROUND

---

### 2.1 Similar Entomology Projects

There is considerable interest in analysing insect trays using computer automated digitisation. Despite being an important topic with considerable interest there are a number of challenges hindering the completion of the task, for example, the variation between insect tray layouts. There are a number of existing systems available which work towards a solution, including analysing or extracting features from insect boxes, such as individual specimens or bar codes.

#### Beyond The Box

A few American institutes with an interest in this area of development have been attempting to find a solution to the insect tray digitisation problem in a recent competition (Beyondthebox.aibs.org, 2016). The challenge for the competition entrants was to produce an imaging system capable of photographing insects from multiple angles and extracting relevant text labels using image analysis.

Considering the difficulty of the photography elements required for the problem to be considered completed, it is not too surprising that no viable solutions were found. However, the project brief has been well thought out and contains some helpful suggestions which may assist for this project with the Museum. A notable idea was the suggestion to use natural language processing for the text extraction. Natural language processing [NLP] involves the digital interpretation of linguistics, such as identifying words or breaking them down into smaller components. This can be applied to optical character recognition to improve the output of detected text based on predicted words. Using an insect-based dictionary would be helpful as a reference for the text analysis in this project.

The institutions involved are continuing to investigate the hurdles and challenging elements found in this project in order to find a way of further constraining the problem in the hopes of finding a solution at a later date. This feedback could be a valuable resource for future system developers looking into this area of research.

#### Inselect

A notable example of digitisation, Inselect (Hudson et al., 2015), produces cropped images of individual specimens from within insect drawer images, assigning metadata to each individual specimen.

The program uses Sobel edge detection as a means of identifying features within an insect tray and selecting connected components to identify as possible specimens. Although the program uses some level of automation in extracting individual components, the segmentation is not perfect and requires manual assistance in order to eliminate false positive matches.

Inselect identifies every individual component and is unable to differentiate between insect and character regions, so the program requires the user to remove non-insect regions manually. It also needs manual separation of connected regions identified as a single insect. This solution would be unsuitable for this project, as the ideal goal for AC-NMW would be to have no user input. A more specialised system could be designed based on constraints identified within the Museum's images to automate this process.

Additionally, the program analyses images individually, and it would be preferable to be able to analyse a set of images at once to create a single XML file for the full contents of the data set.

The program is also integrated with barcode scanning which it uses to obtain additional data for each specific specimen; however Inselect currently has no text extraction capability.

The use of barcodes is particularly reliable in image processing as it is invariant to scale and partial occlusion which is particularly useful in this application, provided that there is a suitable mapping of barcode to specimen data. For future development of digitisation this approach could provide a useful integration method for digital insect trays.

### SatScan® Collection Scanner

Some systems such as Satscan (Blagoderov et al., 2012) focus on optimising the imaging process for insect trays. This system uses specialised camera rigs and lighting in order to photograph high-definition insect tray images. They use a patchwork of small images which are digitally stitched together to produce the most accurate reconstruction possible. This can help to avoid distortion from camera lens shape, light reflection, etc., to maintain high quality detail of the small insects. This is an example of a system designed specifically for this particular application and would be ideal for photographing a museum's insect collection, as the lighting and camera techniques can produce a true to life representation of the tray.

While this may be an optimal method of producing insect tray images, this setup is more advanced than AC-NMW is currently capable of producing, but it may be an interesting idea for AC-NMW to consider for future advancements in the digitisation process.

As the camera setup within the Museum is not of such a refined standard, there may be some problems with the images, for example occlusion of densely packed objects within the trays. The labels are applied to the bottom of the box while the specimens are suspended on pins, therefore, in some trays the angle of specimens further from the centre may cause some overlapping to occur.

## 2.2 Other Related Topics

Other similar investigations are being carried out within the text extraction area of research. The problem of identifying possible character regions within an image has been considered by Epshtein, Ofek and Wexler (2010). The defining features of text regions can be used as a means of finding possible text regions within natural images, such as the continuity of the stroke width for a character region. The insect drawers are of considerably large pixel dimensions due to the high resolution, so conducting a pre-processing step of determining possible regions for text to be identified would increase the efficiency of the program.

Additional optical character recognition techniques have been considered and compared for their uses in identifying text within varying forms of media (Sumathi, 2012). These can be considered as options for suitability with respect to the text extraction for insect drawers.

## 2.3 Relevant Algorithms and Applications

### Edge Detection

Edge detection will be required in order to detect likely locations for characters within an insect tray, significantly reducing the search area. Text detection can be a computationally expensive operation, so reducing the search area will speed up the analysis process. Additionally, this can be used to detect the column boundaries to help interpret insect tray layout.

A number of algorithms are available for edge detection. An important factor to consider is whether the output requires a single pixel-wide edge or a general level of edge strength at each pixel. Examples of these different types include Sobel and Canny edge detection (Das, 2012). While Canny produces a clear response of positive and negative matches for edge

regions, it is sensitive to noise and may omit weaker edges which may be useful for insect tray analysis. Alternatively, Sobel is a much simpler algorithm which detects the edge strength across the entire image. It has the added benefit of finding edge orientation, allowing the search to be limited to vertical or horizontal directionality here applicable.

## Line Detection

The Hough transform algorithm (Duda and Hart, 1972) can be applied to an image after an edge detection algorithm in order to detect lines within the image. Votes are accumulated at each edge pixel detected. For each point a number of lines could be plotted through it at different angles. For each line, the perpendicular intercept through the origin is found to calculate the angle and radius between the origin and line. The corresponding angle and radius are used as coordinates in parametric space. Collinear points will have the same intercepting line, causing an accumulation of votes for this particular angle and radius. Lines in an image can be detected by searching for these peaks in parametric space and mapping the lines back onto Euclidean space.

Although the Hough transform algorithm can be used to detect lines of all angles, the segmentation within insect trays only uses approximately horizontal and vertical lines. Therefore, the algorithm can be simplified by limiting the range of angles detected to approximately horizontal or vertical orientation. This would eliminate possible false positives in Hough space at angles which are not required.

## Text Recognition

For the character recognition algorithm a number of options are available. An idea for a solution to this problem could be to use existing text recognition software. Google has developed a text recognition API (Google, 2016) which can be used to extract text from document photographs. It is capable of interpreting text structure using different region sizes, such as text blocks, lines and individual words.

The museum curators have conducted previous experiments with their drawer images using the Google API, however, the program uses natural language processing and anticipates words from a standard dictionary using Latin based languages. This system struggles with the unusual text of insect and author names. Although this particular design is not suited to the project, the idea of using a dictionary can be applied to the current problem as a way of improving output text. Comparing the detected text to an insect-based dictionary as a reference may help to improve the accuracy of the output data.

Template matching is likely to be a preferable solution to the problem, although the process is often slow and computationally expensive.

There are a number of ways of comparing the template against the image region, for example, zero-mean difference or normalised cross correlation (Docs.adaptive-vision.com, 2017). While a method such as the mean difference - calculated between image pixels and template pixels - can be performed using a simple algorithm, the results are not as reliable as alternative methods with greater computational complexity.

There is likely to be inconsistency in the results of mean difference, as there is a variation in illumination across the tray where boxes or insects cast shadows. Therefore, a normalised algorithm is a better solution as it compensates for the inconsistent brightness in the image.

The efficiency of this more complex algorithm can be improved by preparing some values in advance, such as using a static value for the average brightness of a template character rather than calculating this value at every pixel on the image area. It would also be useful to compute the template matching for multiple characters in parallel, however there would be a

variation in window size due to different dimensions of characters which makes this process less feasible.

### Text Comparison

For comparing strings of characters the edit distance can be used as a measure of accuracy between two similar lines of text. This value can be computed by calculating the number of insertions, deletions and substitutions to change one string into another.

The Levenshtein edit distance (Sulzberger, 2017) is a useful method of analysing the number of character changes, which can be applied to the output strings to compare them with the dictionary references. This gives a numerical value which can be used to more accurately analyse the improvements made during program development.

### 3. SPECIFICATION AND DESIGN

---

The Museum has set up a camera to take standardised high quality images of their insect collection. These images can be used in an image analysis program in order to extract the relevant data to publish alongside the images. As this project will be based on images provided by the Museum there may therefore be time constraints depending on the availability of staff within the entomology department to set up and photograph the insect trays for the project. Consistent features between trays can be used as constraints to apply to the problem, enabling the creation of a specialised product with this application in mind in order to optimise the output of insect tray analysis.

In order to complete the project a number of smaller components will need to be completed. The required features include:

- Column segmentation
- Empty space elimination
- Character detection
- Character to text transformation
- Dictionary keyword matching
- Metadata extraction
- Text to XML transformation

Additional features may assist in the text output process including:

- Specimen detection
- Error correction

#### 3.1 Program Requirements

The initial problem to tackle is interpreting the layout of text and specimens within a drawer. The insects are organised using small boxes which provide the drawer area with segmentation into columns and rows.

The contents are structured similarly to newspaper columns, such that each full column is viewed from left to right. Figure 1 shows an example of a box layout within an insect tray including specimens colour coded to match their relevant species labels.

The label displayed at the top of a box indicates the Genus, or generic name, to which the following insects belong – label 1 – and is often of larger and bolder font. The label indicating the scientific name for a set of specimens can be found beneath the insects – labels 2-4. For insects with higher quantities or larger dimensions, the specimens may span an area across multiple boxes or columns. This consistent structure should help to interpret the tray contents based on relative location within a tray.

Using known features of the image will help to improve the performance of the system, beginning with high level segmentation. The structure of the insect trays consists of a frame containing columns of boxes of insects. The edges of these columns create roughly straight vertical lines which can be used to segment the image. By finding the column edges, the column content between these lines can be analysed in correct structural order.



*Figure 1: Example specimen and label layout of an insect drawer*

The efficiency of image analysis can then be improved by detecting regions of interest within an image. As the species are often organised into individual boxes within a tray, there is a considerable amount of empty space which does not need to be analysed for character matches. As the template matching algorithm may be computationally expensive, eliminating these empty regions will be useful for reducing the search region as much as possible and consequently reduce time required for analysis. This could be done using edge detection or foreground thresholding in order to eliminate featureless regions.

The non-empty regions can additionally be searched in order to find likely candidates for specimens. The background area is relatively uniform so it may be possible to extract non background regions in order to differentiate between insects and text regions. The intensity values along each detected edge can be used to calculate an average to use as a threshold in order to locate foreground and background regions within the tray. This could also be used to find an approximate quantity of specimens for each species to use as additional XML data.

The identified regions of interest can be searched for text matches using optical character recognition. Of the previously considered character detection algorithms, normalised cross correlation is likely to be the most suitable method, and a number of thresholds can be tested to find the optimal value for accurate output. If specimen analysis is successful, the detected specimen regions can be used to detect and remove false positive matches where text regions and specimens overlap.

Additional error correction can be applied by verifying the accuracy of detected text using natural language processing. A dictionary search could be implemented using integration with the Global Biodiversity Information Facility (GBIF) API or external dictionary files.

The API is unlikely to be suitable as a direct source as the formatting may not be consistent over time and the program would require relying on internet access in order to run the search. The Museum has offered alternative dictionary resources for string matching. These data would be in the form of excel spreadsheets, which would require converting into another format to use within the program.

There are a number of approaches which could be used, for example using SQL database integration or JSON/XML files. Considering the speed and simplicity of access to data, JSON may be preferable to making SQL queries for each string. Using JSON allows a reduction in redundancy by structuring the data beneath the same parent value, rather than needing to search for a parent value for each species in table format.

The dictionary can be used as a reference for output text. The detected text and dictionary keywords can be compared using the Levenshtein edit distance as a measure of accuracy. A very low value for the edit distance will indicate a probable match where the dictionary reference should be used to replace the text found.

These data can then be converted to XML by surrounding text elements with '<keyword>' tags according to the format requested by the Museum. The image metadata can be used as an additional source of information to include in the XML output. The image name and capture date can be used as specific identifiers for the image to help associate data correctly after the analysis is completed. The capture date may also help differentiate between versions of the data for the same insect tray. These data can then be exported to an XML file.

### 3.2 Program Structure

The required program components can be combined into a single program to process each image individually, extracting specimen data and detecting text for each before compiling results into an XML file. Some of the processes will produce output data that will be needed for later analysis processes such as the identified columns or regions of interest. A possible design for this system can be seen in figure 2.

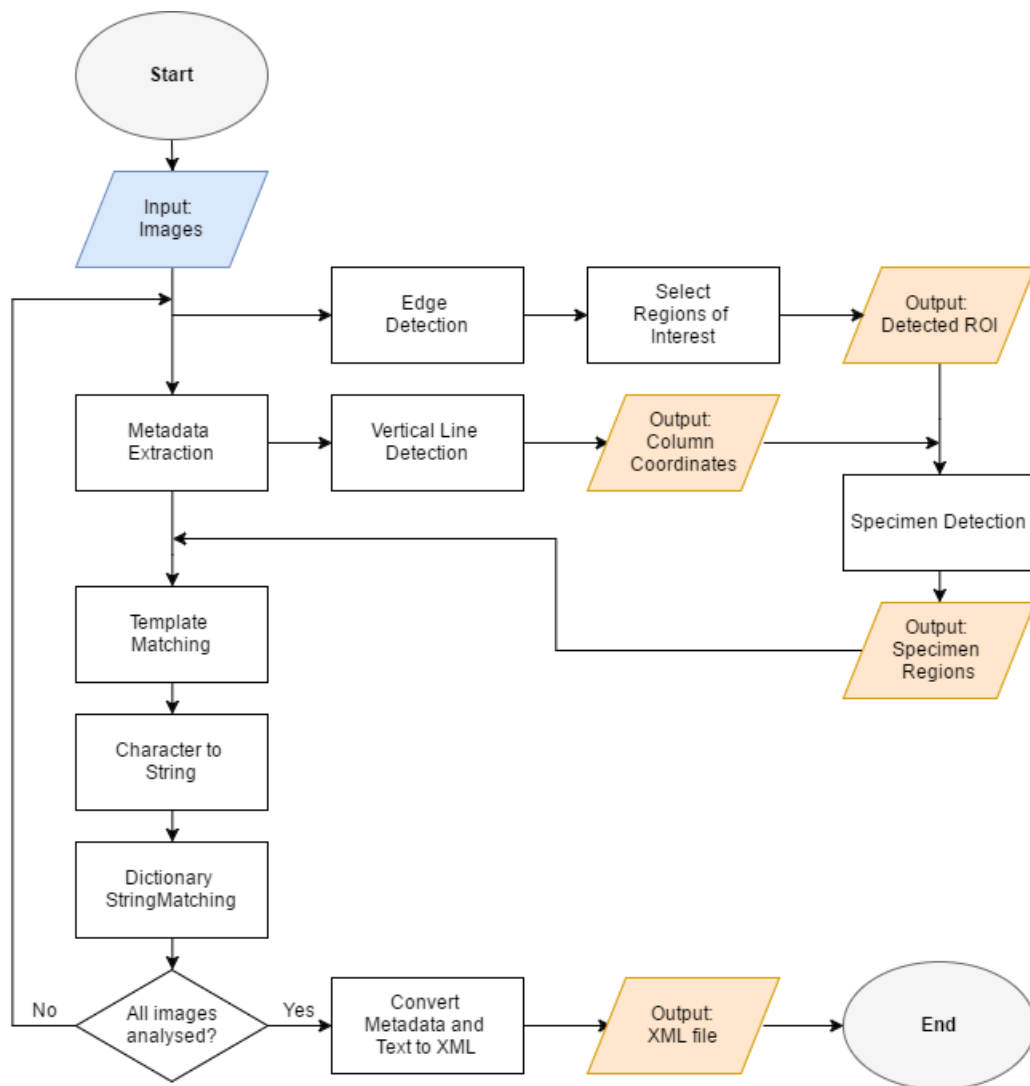


Figure 2: Flow chart displaying program design

## 4. IMPLEMENTATION

---

### 4.1 Column Segmentation

Initial development requires column segmentation to analyse the image sections in the correct order.

As the column edges may not be perfectly vertical, the optimal column edge can be detected using a small range of angles. The Hough transform algorithm detects lines in an image by accumulating votes using the edges detected in the image. This requires an edge detection algorithm before the Hough transform can be applied. The Sobel edge detection is a preferred choice as it detects edges with directionality and a level of edge strength. The Sobel algorithm can be applied with only the vertical edges being detected to improve the output of line detection.

Although at first glance the lines appear obvious to the human eye, there is a weaker contrast between the different tones of shadow between boxes than the black and white text boundaries. This means that the threshold to maintain column edges requires a low value to ensure that the edge regions are successfully detected. This prevents a large number of additional edge regions from being omitted, which may decrease the reliability of peaks in Hough space.

This problem can be improved by thresholding using the proportion of edges for each vertical column of pixels across the image. Although the box edges may not be perfectly aligned, there should still be a significant increase in weak edge regions near a box edge by comparison with the blank background areas inside columns.

The Hough transform is applied to the detected edges to accumulate votes for possible lines. The angles can be limited to a range of  $10^\circ$  either side of vertical to exclusively locate vertical lines. At each edge pixel the value of  $\rho$  (rho) is calculated at each angle  $\theta$  (theta) and the corresponding point in Hough space incremented.

```
for(int pix = 0; pix < totalPix; pix++){
    theta = 80;
    for(double angleT = -Math.PI/2+(80*Math.PI/180);
        angleT < Math.PI/2-(80*Math.PI/180);
        angleT += Math.PI/180)
    {
        //rho = xCos(theta)+ySin(theta)
        rho = (int) Math.round(xPoints[pix]*Math.cos(angleT)
            + yPoints[pix]*Math.sin(angleT));
        houghMatrix[rho+maxRho][theta]++;

        theta++;
    }
}
```

*Figure 3: Hough transform code used for line detection*

After applying the Hough transform algorithm, the parameters can be mapped back to an approximate x-coordinate for each line detected. The optimal lines found can be used as column boundaries during text extraction so that each column can be processed individually.

### 4.2 Selecting Regions of Interest

The edge detection output from the previous analysis can be used to reduce the search regions for character detection by limiting the search to regions which are not empty. The text regions

will have a strong edge due to the contrast between black ink and white paper. The strongest edges can be selected to find possible locations of text characters within the insect drawer.

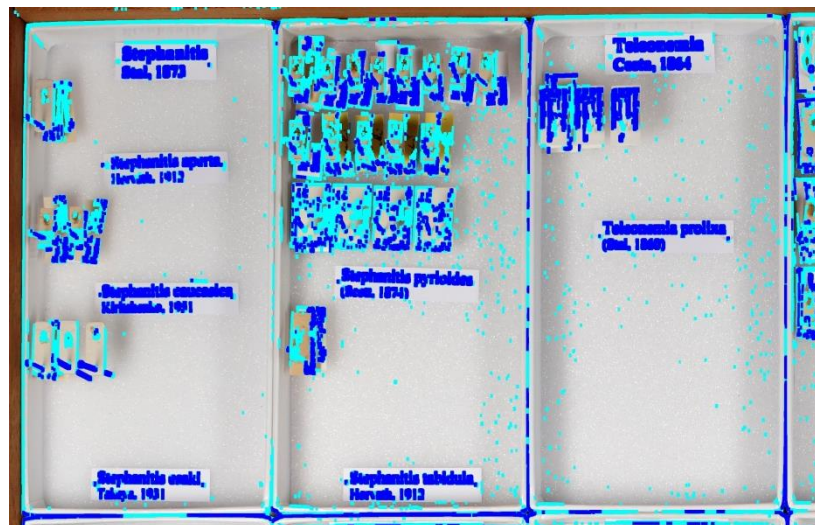


Figure 4: Identification of weak edge (light blue) and strong edge (dark blue) regions

The edges can be selected at different thresholds for different purposes in the remaining image analysis process. A high threshold can be used to identify character regions and an additional lower threshold for specimens.

These edges will be relatively sparse, so they can be expanded to cover a larger area surrounding the detected edges. This will allow a simple verification during template matching, by checking the regions of interest for a positive match within the character region before applying the template.

Searching the entire area could be computationally expensive, so a limited search would be preferable. Initially the centre pixel within a template was used to check for a plausible search region, but a number of characters were failing to find matches due to a lack of features at the centre, such as the letters 'C' or 'O'. Instead the character can be searched for a positive match at a quarter of the character's height up or down along the centre line, as this is more suitable for the full range of character shapes.

### 4.3 Specimen Segmentation

Using the previously detected edges can also help to find an approximate threshold for segmenting the foreground and background regions by averaging the image pixel values at each edge pixel. Applying this threshold to the image produces a segmentation of foreground and background regions.



Figure 5: Specimen segmentation - basic thresholding

While it seemed feasible that thresholding the image would be likely to segment individual specimens against the background colour this also segmented particularly dark shadow regions (Figure 5). These shadows were often created by the additional labels pinned beneath smaller specimens with a slight elevation from the bottom of the box.

Before analysing a component it must be identified by locating the connected components for this shape. A recursive method can be used to identify connected components by labelling each connected pixel with a matching identifier. However, as the images are high-resolution, the components contain a large number of pixels which is likely to exceed the stack size in a recursive method. This can be reduced by eliminating all but the outermost edges of each shape, removing any pixels which are completely surrounded by foreground regions, e.g. centre of Figure 6, example 2. A recursive search can then be applied to label all connected components, removing any shapes which are either too small or too large.

When considering how to eliminate these additional detected shapes, the rectilinearity can be a useful feature to detect as a method of differentiating between specimens and shadow regions.

The straight lines along shadow boundaries tend to lie on an approximately vertical line, therefore, using polygonal approximation (Grigore and Veltkamp, 2003) can help to determine whether a component contains a vertical edge. As the search would be for straight vertical lines, the line for analysis can be generated using the leftmost or rightmost pixels along the vertical length of the shape.

The polygonal approximation can be conducted by recursively analysing line components. The expected straight line between the two end points is used to find the point of maximal error on the curve detected. If the error value is above a threshold distance, the line can be split into two halves at this point, before repeating the process on each half of the line. When a line is below the error threshold, if the length and gradient resemble a vertical line, this connected component can be eliminated as a specimen candidate.

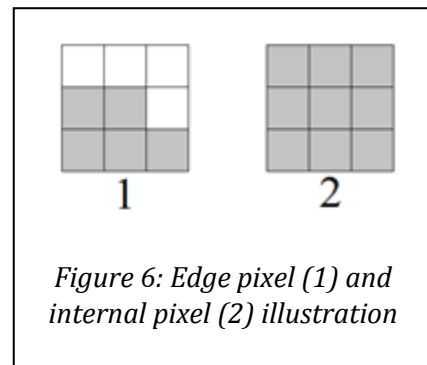


Figure 6: Edge pixel (1) and internal pixel (2) illustration

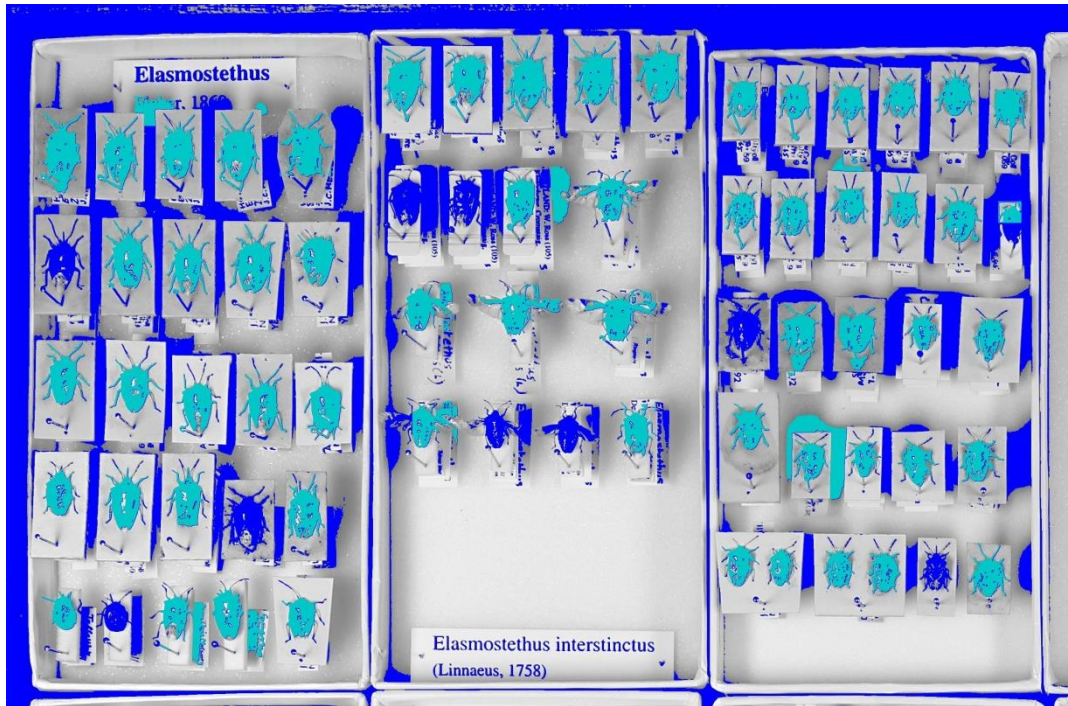


Figure 7: Specimen segmentation - removal of linear components using polygonal approximation

Any small regions such as text characters could be eliminated by analysing the ratio between edge pixels and internal pixels. The previously removed regions inside the component area can be counted and compared with the number of edge pixels. The characters tend to have consistent, narrow stroke width, giving them a much larger edge to area ratio, while the insects tend to have a much smaller outline proportion by comparison.

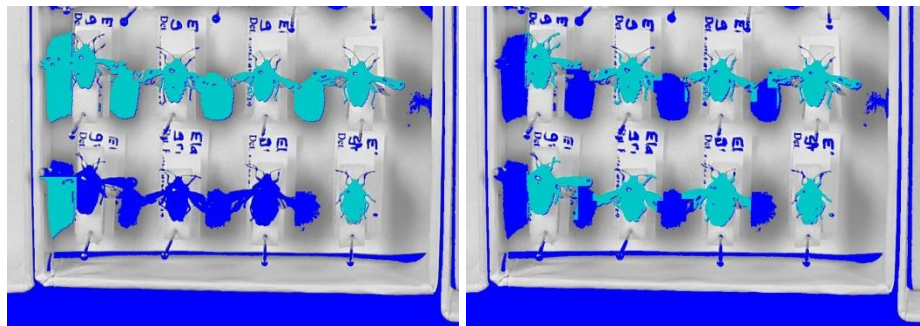


Figure 8: Specimen segmentation - before and after intersection with detected edges

Although these steps create a significant improvement, using the intensity value as a segmentation method creates additional component edges along shadow boundaries which would not otherwise be detected as an edge. This can be improved using the regions of interest to eliminate specimen boundaries which were not detected by the Sobel edge detection algorithm. The intersection of the intensity threshold and edge detection should produce a more reliable component edge.

Finally, a number of cases occurred where antennae or limbs of small insects overlapped the edge of the labels beneath them, causing them to be eliminated as an extension of the shadow region. These extremities were removed by detecting components below a minimal width and removing them before performing the additional steps. This allowed the components to be analysed individually, successfully detecting a larger proportion of insects.

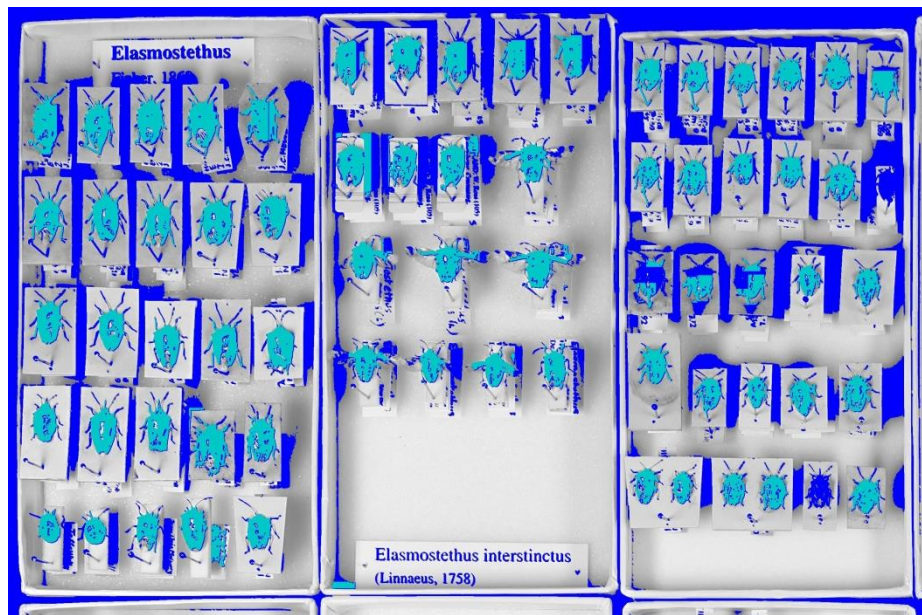


Figure 9: Specimen selection - sparse region removal

#### 4.4 Character Recognition

The character recognition task required a significant amount of time to complete which was not anticipated in the original project plan. The algorithm itself was simple to produce, however a number of additional problems were encountered during this process.

When applying the normalised cross correlation to the images originally provided by the Museum there were a significant proportion of errors in almost all strings detected. Although the images seemed to be high resolution, upon closer inspection the characters were relatively pixelated and the images had visible artefacts, likely caused by jpeg compression.

The entomology department later produced some higher resolution images to combat this issue; however, significant time was lost producing character templates and testing configurations for character recognition at multiple resolutions across the range of font sizes used in the trays.

After a more successful outcome with the higher resolution images, there were additional problems to address. A number of substitutions occurred during analysis where characters of larger dimensions were being incorrectly replaced, for example the letter 'm' being substituted for a similar set of characters such as 'rn'. As the larger characters cover a greater area it is likely to contain a larger proportion of errors than a similar character with smaller area. Ideally this could be improved using an error correction process either during template matching or in post processing.

Similarly the algorithm would detect additional characters within the area of a larger character, such as an 'i' at the edge of a character 'm'. The search was designed to check for previously discovered matches within similar dimensions to the current character size before storing the best match at the optimal location for this character. The dimension-based search was designed to prevent double letters from being detected as a single character. However, the limited search region may not find the current optimal match if the previous character is not close enough to the search region, detecting additional characters within existing character regions. This was improved by adding additional data points across the width of the character in order to ensure that the current best match is found within the search region for smaller characters.

Other incorrect characters substitutions could be improved by simply modifying the template used for characters producing false positives. This helped to prevent cases where a positive match was found as a subsection of another character, for example a 'y' being detected

as the letter 'v'. By extending the height of the character 'v' to include the tail of the 'y' and modifying other similar characters to cover the same dimensional area the number of false positives was significantly reduced.

#### 4.5 String Output

When converting the detected characters into strings of text, the character heights can be used to find the set of characters which appear to be on the same line. Although it should be possible to detect lines of text based on an empty row between rows of characters, some detected characters may be too close in vertical proximity to leave a full empty row of pixels between them. Therefore, based on the structure of text, the lines can be detected by finding the widest point at which the number of character regions is the maximal value in a row. The widest point should be found at the bottom of the line of text, just above the tails of letters such as 'y' or 'g'. This allows the text to be extracted even if the following line of characters partially overlaps, as seen in figure 10, example 2.

This may be caused by false positives along box edges resembling vertical line shaped characters such as 'I' or 'l'. False positive matches will not have any relation to the text region and may interfere with the output. More evaluation of the image could be done in future to remove these lines before applying the character detection to improve the text region selection process.

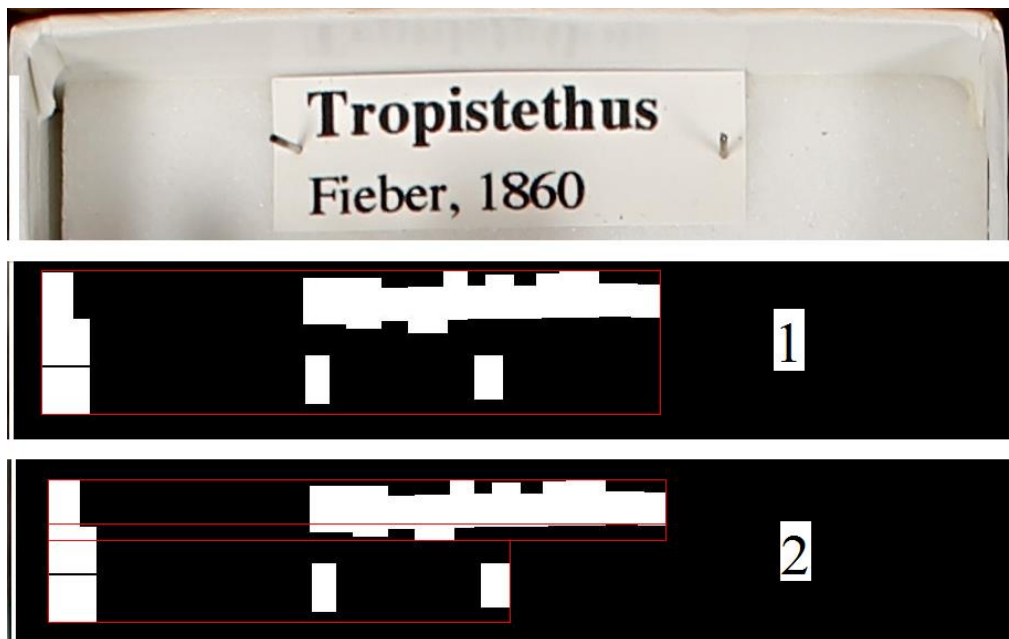


Figure 10: Segmenting detected text regions using empty rows (1) or decrease in row width (2)

#### 4.6 Dictionary Matching

The JSON integration could be implemented using the JSON simple java library (Noakes, 2017) in order to integrate JSON arrays and objects. An additional program was created in order to convert the excel spreadsheet data into JSON. This required reading the excel files using Apache POI library for Microsoft documents (Apache, 2017).

```

{
  "Genus": [
    {
      "Name": "Oestridae",
      "Species": [
        {
          "Name": "Cuterebraneomexicana",
          "Author": "Sabrosky, 1986"
        },
        {
          "Name": "Cobboldia elephantis",
          "Author": "(Cobbold, 1882)"
        },
        {
          ...
        }
      ],
      "SpeciesCount": 27935
    },
    {
      "Name": ...
    }
  ],
  "GenusCount": 1032
}

```

*Figure 11: Example JSON insect dictionary structure*

The initial development was based on the GBIF dictionary, but as a number of species were missing, the dictionary collection was expanded to use the museum's data and the extended GBIF collection. Using these dictionaries, the detected text can be compared with dictionary keywords using Levenshtein edit distance. The Apache StringUtils library (Apache, 2016) contains a simple Levenshtein comparison method which can be used to find the best match.

Considering the additional constraints which can be applied to the problem, the layout and font size could be used as further improvement on the dictionary matching method. Some of the labels are printed using different font size as an indicator of relationship between adjacent lines, such as an insect's scientific name having a larger font size than its author. If a line of text has been found to have greater font size than the following line, the dictionary can be searched for a set of data – name and author – for the adjacent lines rather to improve the accuracy. If there is no significant difference between the strings and the dictionary references it will help to validate the pair of strings together.

The general name also required a specific case as the font size of the author would also be greater than the font size of the following species. This case can be avoided by checking for relationship between more than two adjacent lines.

On further discussion with museum curators, an additional idea developed to use the 'order' of an insect to increase the reliability of dictionary searching. The image metadata can be edited during the photography process in order to give additional information to assist with dictionary matching. The 'order' pertains to a subsection of insects which can be used to reduce the set of data for keyword matching. This provides a further level of validation, preventing the algorithm from incorrectly finding a good match from the wrong order of insects.

#### 4.7 Metadata Extraction and XML Output

The metadata was a relatively simple process which could be completed with the help of Metadata Extractor library (Fang, 2017). This required a search for the image 'Create Date' and 'Subject' to include in the XML output.

The previously mentioned 'order' data added by the entomology department was included within the 'Subject' tag. This can be included as an additional keyword in the XML output under an '<order>' tag. This could optionally be made available to collection viewers to help check that the data is accurate.

The output text could then be transformed into XML by wrapping detected text in keyword tags. For cases where the dictionary text was used as a replacement for the previously detected string the source of this data can be added as additional information, for example, '`<keyword source="GBIF">`'.

## 5 RESULTS AND EVALUATION

A number of values within the program can be modified in order to optimise the performance. The set of images provided includes nine high-resolution images with printed insect labels. The best results will be found by using as many training images as possible to configure the settings; therefore the full set of images should be used while modifying settings to measure the optimal conditions.

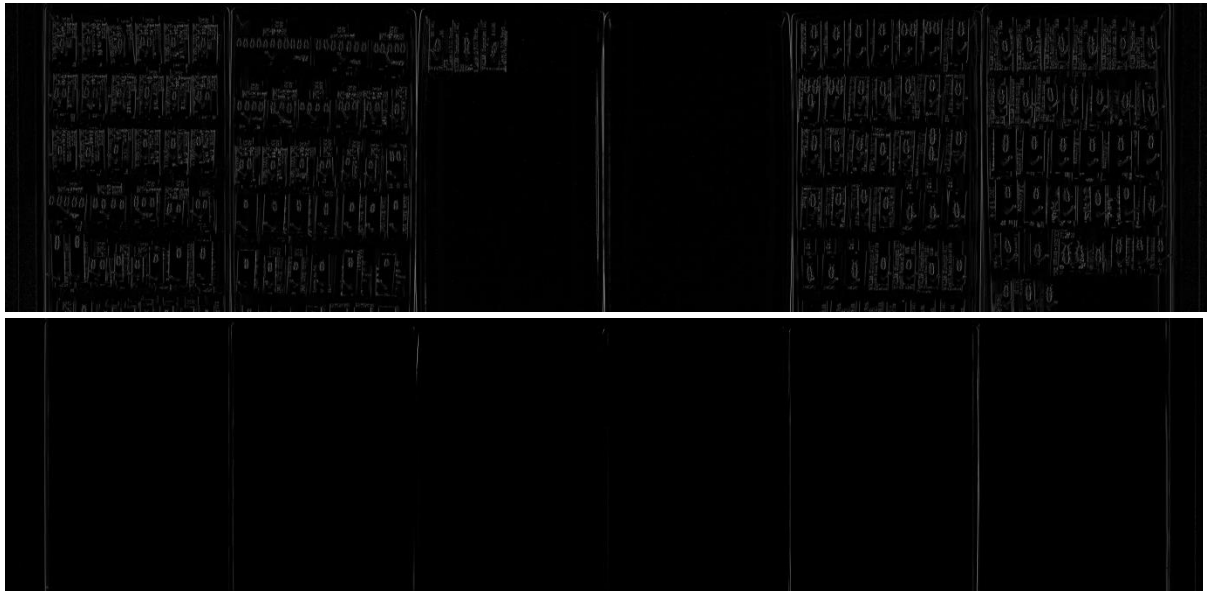
<b>Number of Images (with printed labels)</b>	9
<b>Single Image Width/Height</b>	5792
<b>Total Image Characters (including spaces)</b>	5918
<b>Total Insect Specimens</b>	3200

*Table 1: Program testing high-resolution image set*

### 5.1 Hough transform

For the line detection algorithm the number of columns detected depends on the variables chosen. There is a variation in detected edge output depending on the variables used for Sobel edge thresholding, weak edge quantity and Hough space threshold. The values chosen will need to prevent producing fewer than the required number of column edges as this will incorrectly merge two column regions. Additionally, lines intersecting column content should be prevented as this will incorrectly split the column data.

The percentage of vertically aligned weak edges can be used as a method of removing column content without eliminating column edges. This method can be tested by producing an output image of the removed regions before and after thresholding at different percentages and visually assessing the outcomes.



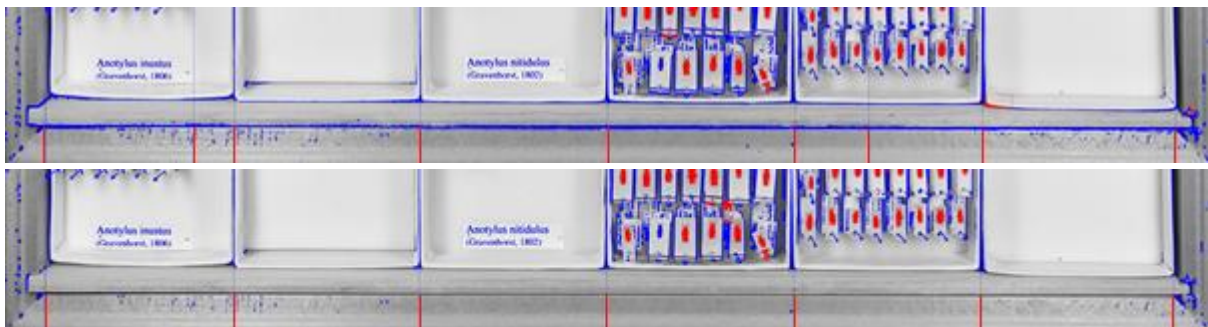
*Figure 12: Sobel edge detection output, before and after thresholding by percentage of vertically aligned edges*

The program can then be modified to output a value for the number of columns detected and an image with the detected lines drawn onto the image content. These can be used to measure the accuracy of Hough transform performance by visually checking the output of the algorithm.

A reasonable method of selecting the Hough threshold is to use the maximum value in Hough space as a starting point. The Hough threshold value will therefore use a decimal number and multiply this value with the maximum Hough value. Similarly, the Sobel edge detection can be scaled up to 255 to display grayscale output during testing. Therefore the threshold will be in the range of 0 to 255.

Sobel Edge Threshold	Hough Space Threshold Multiplier	Average Number of Lines Detected	Average False Positive or Negative Matches
80	0.40	6.2	0.8
60	0.40	7	0
40	0.80	2.6	4.4
40	0.50	7.5	0
40	0.40	8	0
40	0.20	12	2.5

*Table 2: Sobel and Hough threshold value analysis for images with seven column boundaries*



*Figure 13: Oversegmentation vs accurate segmentation of column boundaries*

Although the number of columns should be at least equal to the number expected from visual identification, there are multiple box shadows along the column edges which can produce multiple positives. This may not necessarily be an incorrect match, as this could be helpful later in the program. These shadows often produce false positives when compared with letters that resemble straight vertical lines. Maximising the number of lines may improve the performance by preventing the character detection algorithm from searching these regions, however the risk of false positives between column boundaries should be minimised to prevent splitting text regions. Therefore, the Sobel threshold of 40 and Hough threshold of 0.5 times the maximum Hough value will be used as a good match for Hough transform output.

## 5.2 Template Matching

The template matching algorithm needs to be tested in order to find the optimal threshold for accurate character detection. The result of the normalised cross correlation (NCC) algorithm will be a value between 1 and -1, with 1 being a perfect match and -1 as the exact opposite. The image is extremely unlikely to contain exact matches, therefore the NCC threshold will be a value lower than 1.

The algorithm is initiated with a sparse search of one pixel in every 2x2 square until it encounters a pixel above the NCC threshold value. A more thorough search is conducted within this region of the image to locate the optimal character match at this point.

The best value for the threshold can be determined by using the Levenshtein edit distance to compare the output text at a number of thresholds to the correct label text from manual extraction. As the image set is limited the most accurate data can be produced by using all insect tray images. The character detection algorithm can be applied to the images and the results of text extraction saved as text files with the corresponding tray identifier. Similarly, the

labels can be copied out manually into text files with as much accuracy as possible to use as a comparison with the detected text. The Levenshtein edit distance can be performed on these text files in an external program in order to compare the expected and detected text.

Threshold Value	Total Edit Distance	Average Error Percentage
0.68	1228	20.8
0.69	1197	20.2
0.70	1180	19.9
0.71	1234	20.9

*Table 3: Levenshtein edit distance results for all insect trays (5918 characters)*

Due to the complexity of the program the analysis takes a significant amount of time to complete. Although more data would be desirable, the program requires a number of hours to run to completion on the full set of images and limited time was available during the project.

The investigation suggests that the optimal threshold is around 0.70, which can be used as a starting point for template matching. However this still contains around 20% error occurrences which could be further improved using specific error correction based on frequent errors. This would require additional analysis of the character substitution rates to find a suitable correction method.

### 5.3 Dictionary Matching

Although it should be a simple process to match the text to the dictionary references there were often incidents where an insect was not found in the dictionary. This also occurred for labels which contain a higher hierarchy level such as the general species rather than a specific scientific name. Therefore the initial analysis had to be conducted with manually copied text added to the dictionary in order to check the functionality of the string matching process.

When comparing the expected output text with the dictionary the result was 100% accurate, however given the number of inaccuracies within the detected text and the similarities of keywords in the dictionary it is difficult to analyse this component. However, it performs as expected to match text to dictionary references, despite differences in formatting in dictionary references, therefore this has been successfully completed.

When comparing the output strings to the dictionary, considering that it is possible that the insect data will be missing, it would be unwise to replace every string with the closest match in the database. A percentage of the string is used as a threshold for which the string should not be replaced by the dictionary value. If more than half of the string length is incorrect then this is not a good enough match and no substitution should occur.

### 5.4 Specimen Detection

When performing the specimen detection, the incremental changes discussed during the implementation section can be applied to all images to measure the improvements that each component has on the detection results.

Algorithms Applied	Detected Specimens	Detection Percentage
Basic Thresholding	127	42
Polygonal Approximation	164	55
Edge Detection Comparison	252	84
Sparse Region Removal	313	105

*Table 4: Specimen segmentation improvements applied to example image with 298 specimens*

Image	Number of Specimens	Detected Specimens	False Positives	Accurate Matches	Accuracy Percentage	Detection Percentage
018617	599	553	114	439	79	73
018618	427	374	55	319	85	75
018619	298	315	26	289	92	97
018620	128	51	11	40	78	31
018621	431	254	43	211	83	49
018622	562	131	120	11	8	2
018623	385	91	34	57	63	15
018624	152	134	14	120	90	79
018625	218	187	47	140	75	64
<b>Average</b>	<b>3200</b>	<b>2090</b>	<b>460</b>	<b>1630</b>	<b>78</b>	<b>51</b>

*Table 5: Accuracy based results of specimen region extraction*

The output from the final specimen detection analysis has significant variation in percentage of detected insect which is often in images which include insects much larger or smaller than the average specimen size. The detection algorithm uses a number of processes which address the specific problems encountered for medium sized insects but has not been extended to the full range of specimens. However, of the regions identified as specimen candidates there is a reasonably high accuracy rate.

Although it would be ideal to locate every insect, it is preferable to have sub-optimal identification than to falsely identify a character region as an insect. If the specimen regions are used to eliminate false positive text matches where these regions overlap, identifying a character as an insect could cause the program to throw away correct data. Unfortunately this means that the solution is likely to be over-constrained and will not find insects with similar properties to text characters, especially smaller insects.

Given more time the analysis process could be automated by using image processing techniques to verify false positives or false negatives. A set of test images could be produced by manually indicating the insect regions and comparing this to the regions found during specimen detection.

## 5.5 Additional Evaluation

A number of additional components were required for the successful implementation of the resulting program.

The external program designed for the conversion of Excel to JSON was able to produce a suitable JSON output which could be used within the final program as an additional verification method. This additional program could be helpful in future if there is a desire to produce additional updated dictionaries in the same format for an extension of the dictionary matching. Furthermore, the discussion of ideas with museum curators helped to add another level of reliability by filtering the dictionary created using the 'order' included in metadata.

The template set required as a resource for the template matching algorithm was produced for each of the required font sizes and weights. These templates were improved where possible to optimise the algorithm output based on logical problems encountered such as character similarities.

The image edge detection process was implemented successfully using the Sobel edge detection and was used to identify likely character regions within the insect drawer images. Considering that the program takes multiple hours to complete using only the identified regions of interest, the program would be considerably slower without this component included within the system. The limitation of empty regions significantly decreases the search space which would otherwise take much longer to process due to the high-resolution of the insect drawer images.

## 6 FUTURE WORK

---

There is a wide scope of development in this topic which can be explored in order to help work towards optimising the digitisation of museum collections.

For the tasks completed during this project, a number of improvements could be made such as an additional method of error correction for detected label text. Further layout analysis could help to match the number of specimens between identified labels with the correct species. This specimen output could also be used

The Museum also has a wide variety of specimen drawers with varying layouts which have less consistent structure or labels which exceed the boundaries of column edges. This would provide an additional level of complexity. Testing the program on these images would be unlikely to produce a good output as it has been designed around specific constraints of the image set provided. Further development could be continued in order to create a generic solution for the entirety of their insect tray structures without relying on column boundaries.

Additionally, there are a number of trays which use handwriting which would be a bigger challenge to tackle. The template matching algorithm was tested with a set of handwritten letters, but no useful output could be created. It appears as though the characters use a relatively uniform lettering, but the slight variations in angle and size can easily affect the output of character detection. Given the limited time available during this project no solutions for the handwritten text could be found. However there are some possibilities for analysis of handwritten trays, such as distance transform for binary character templates as an alternative method to template matching.

The problems encountered and useful constraints found to improve the output can be considered in future projects as a basis for a more robust method of text extraction.

## 7 CONCLUSION

---

This research was conducted as a pilot project for the digitisation process for the insect collection at the National Museum Wales, Cardiff. Similar investigations have been undertaken by other museums and institutions which provide a valuable source of information on how to begin looking into digitisation. These have helped as an introduction to the topic of entomology based digitisation and provided valuable insight into the problems encountered.

While there are a number of ongoing investigations and similar programs in image text extraction and recognition, often these are more suitable for generalised applications such as detecting text within documents with limited non-character features. While they address the same issue they do not perform as expected when applied to a more unusual image. Therefore, creating a program suited to the specific insect drawer structure used at the Museum could make improvements on these similar projects.

The insect drawer analysis can be more successfully performed by maximising the constraints which can be applied to the problem. Understanding the drawer layout helps interpretation by breaking down the analysis task into a series of processes. From high level segmentation of column regions down to the identification of specimens, the features of the image can be used to help create a suitable method of interpreting the contents of the insect drawers.

The identification of specimens within the program aims to make improvements on the partially manual identification found in Inselect. The solution addresses problems such as avoiding character regions and separating connected specimens using the features of the image contents. Although there is still work to be done to improve the overall success rate, the prevention of falsely detected character regions is a preferable outcome.

The tasks performed during pre-processing, such as eliminating empty spaces and identifying regions of interest are extremely helpful in improving the performance of algorithms and reducing the time required for analysis. High-resolution image processing is an intensive task which can take a considerable amount of time and computer memory to complete.

The use of the dictionary with the inclusion of specific terminology relevant to entomology and based on the museum's own data, together with the addition of the 'order' of an insect aims to increase the reliability of dictionary searching. The ability to integrate the system with dictionary referencing can produce much more reliable results, and the use of Levenshtein edit distance to compare the output text was found to be effective as a measure of similarity for dictionary corrections.

The components of the resultant program address some of the issues found in insect drawer digitisation; however the vast scope makes it difficult to completely solve the problem. Although a significant amount of work is required to improve the output of text extraction, the results of this investigation indicate that there is a strong possibility of producing a reasonable output given a suitable error correction method.

The results of this project make a promising start towards digitising the insect tray drawers. This can be used as a basis for the continuation of the digitisation process and the Museum has made plans to continue development in this area of research to work towards a full solution which can automate the digitisation of their entomology collection.

## 8 REFLECTION ON LEARNING

---

Digitisation is an extremely exciting topic with a huge scope for development and has provided an interesting and rewarding topic for this project.

Throughout the task there have been a number of challenges, for example the unexpected complexity of the segmentation of specimens within an insect tray, which required taking a step backwards in order to find a solution. Although I was able to identify the possibility of using straight vertical edges to differentiate between specimen and non-specimen regions, knowing how to solve this in a program is a completely different task. Using the topics discussed in Visual Computing helped to simplify the problem by applying my knowledge of polygonal approximation of curves.

I have also learned the value of knowing when to stop investing time into a problem which cannot be fixed through programming. When initially developing the template matching algorithm the images were not of high enough quality to produce a reasonable output. This problem could be fixed at a higher level by producing better quality images rather than attempting to improve the algorithm. Although this produces additional work for the client, it is better to produce a higher quality solution to the problem given than to waste time on an outcome which is sub optimal.

I found that taking on a project of this scale alone is a daunting task and knowing when to ask for help or for feedback on my ideas has allowed me to maintain consistent development each week and increase the quality of my work. A number of skills can be learned through discussing ideas with someone who has more experience in the topic, and I found that discussing my ideas with my supervisor gave me confidence that I was developing my solution in a logical direction.

Being able to apply my problem solving skills to this task has given a great sense of achievement and I hope that the digitisation process within the Museum will continue to develop.

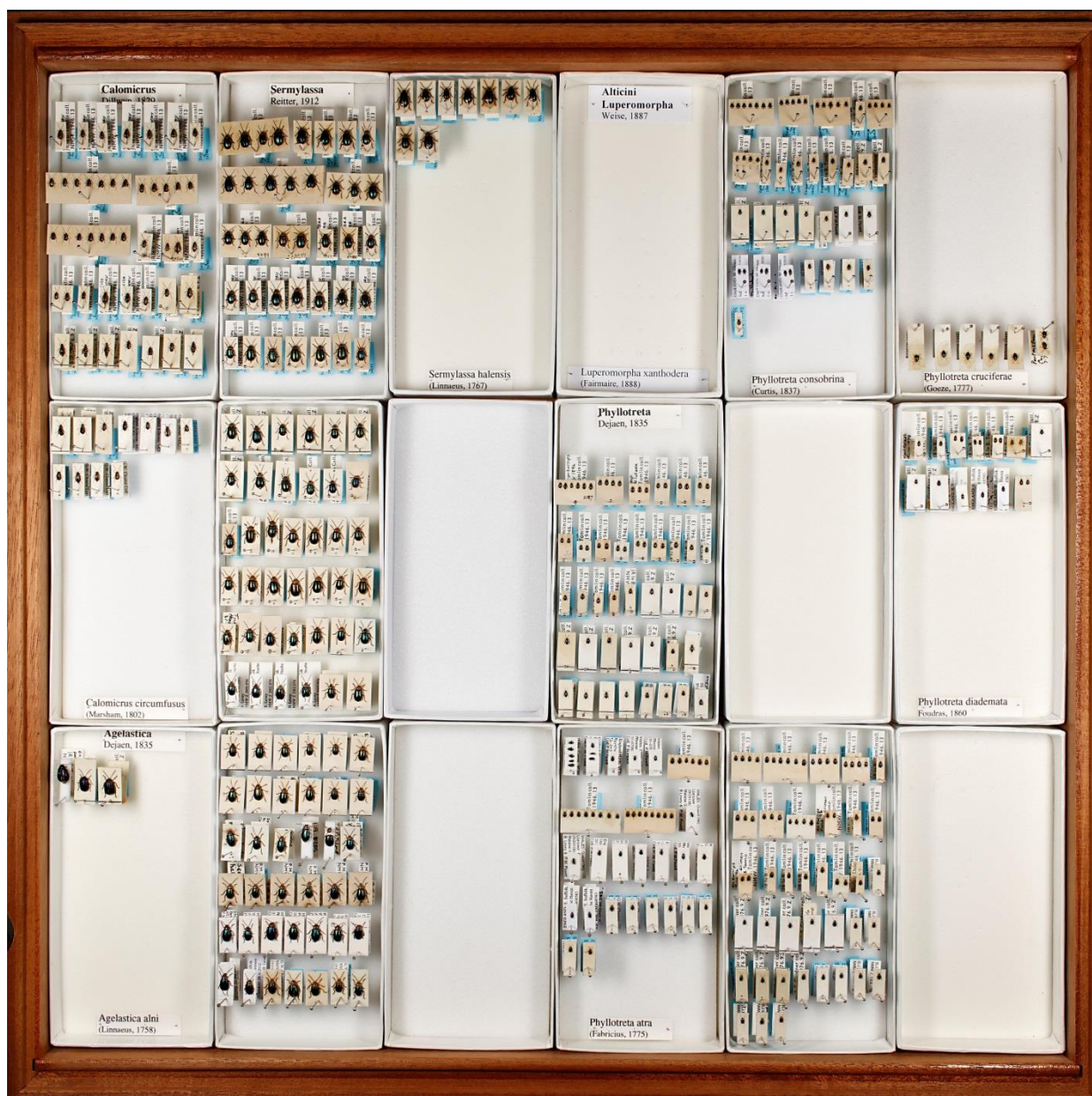
## APPENDICES



















## REFERENCES

---

- Apache. (2016). *Lang – Home*. [online] Commons.apache.org. Available at: <https://commons.apache.org/proper/commons-lang/> [Accessed 5 May 2017].
- Apache. (2017). *POI-HSSF and POI-XSSF - Java API To Access Microsoft Excel Format Files*. [online] Available at: <https://poi.apache.org/spreadsheet/> [Accessed 5 May 2017].
- Beyondthebox.aibs.org. (2017). *Beyond The Box / Overview*. [online] Available at: <https://beyondthebox.aibs.org/overview.html> [Accessed 5 May 2017].
- Blagoderov, V., Kitching, I., Livermore, L., Simonsen, T. and Smith, V. (2012). No specimen left behind: industrial scale digitization of natural history collections. *ZooKeys*, 209, pp.133-146. doi: 10.3897/zookeys.209.3178
- Chowdhury, G. (2005). Natural language processing. *Annual Review of Information Science and Technology*, 37(1), pp.51-89. doi: 10.1002/aris.1440370103
- Das, S. (2016). Comparison of Various Edge Detection Technique. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 9(2), pp.143-158. doi: 10.14257/ijsp.2016.9.2.13
- Docs.adaptive-vision.com. (2017). *Template Matching*. [online] Available at: [http://docs.adaptive-vision.com/4.7/studio/machine\\_vision\\_guide/TemplateMatching.html](http://docs.adaptive-vision.com/4.7/studio/machine_vision_guide/TemplateMatching.html) [Accessed 5 May 2017].
- Duda, R. and Hart, P. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), pp.11-15. doi: 10.1145/361237.361242
- Epshtein, B., Ofek, E. and Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/cvpr.2010.5540041
- Fang, Y. (2017). *fangyidong/json-simple: A simple Java toolkit for JSON.* [online] GitHub. Available at: <https://github.com/fangyidong/json-simple> [Accessed 5 May 2017].
- Google. (2016). *Text Recognition API Overview / Mobile Vision / Google Developers*. [online] Available at: <https://developers.google.com/vision/text-overview> [Accessed 5 May 2017].
- Grigore, O. and Veltkamp, R.C., 2003. On the implementation of polygonal approximation algorithms. *Department of Information and Computing Sciences, Utrecht University, Tech. Rep. UU-CS-2003-005*.
- Hudson, L., Blagoderov, V., Heaton, A., Holtzhausen, P., Livermore, L., Price, B., van der Walt, S. and Smith, V. (2015). Insect: Automating the Digitization of Natural History Collections. *PLOS ONE*, 10(11), p.e0143402. doi: 10.1371/journal.pone.0143402
- Mantle, B., LaSalle, J. and Fisher, N. (2012). Whole-drawer imaging for digital management and curation of a large entomological collection. *ZooKeys*, 209, pp.147-163. doi: 10.3897/zookeys.209.3169
- Noakes, D. (2017). *drewnoakes/metadata-extractor: Extracts Exif, IPTC, XMP, ICC and other metadata from image files*. [online] GitHub. Available at: <https://github.com/drewnoakes/metadata-extractor> [Accessed 5 May 2017].

- Olsen, E. (2015). *Museum Specimens Find New Life Online*. [online] Nytimes.com. Available at: <https://www.nytimes.com/2015/10/20/science/putting-museums-samples-of-life-on-the-internet.html> [Accessed 5 May 2017].
- Sulzberger, C. (2017). *Efficient Implementation of the Levenshtein-Algorithm, Fault-tolerant Search Technology, Error-tolerant Search Technologies*. [online] Levenshtein.net. Available at: <http://www.levenshtein.net/index.html> [Accessed 5 May 2017].
- Sumathi, C. (2012). A Survey On Various Approaches Of Text Extraction In Images. *International Journal of Computer Science & Engineering Survey*, 3(4), pp.27-42. doi: 10.5121/ijcses.2012.3403