Towards Automated Transcription of Label Text from Pinned Insect Collections

Nitin Agarwal
University of California, Irvine
agarwal@ics.uci.edu

Nicola Ferrier Mark Hereld
Argonne National Laboratory
{nferrier, hereld}@anl.gov

Abstract

We present a computer vision system that can transcribe the text on tiny printed labels stacked beneath pinned insects (as found in museum collections). The approach uses multiple views of each label because the labels are often occluded by the pin, the insect specimen, and other labels. Our approach handles occlusion and the extreme viewing angles required to image the stacked labels. Automated image analysis identifies the lines of text and then aligns and rectifies the images. Combining the aligned and rectified images from multiple viewpoints enables us to create a composite image that can be read using optical character recognition tools (OCR) to extract the text. We provide experimental demonstration using both museum specimens and experimental test labels.

1. Introduction

Large-scale collections of pinned insects provide data for studies of taxonomy, biodiversity, biological conservation, land management, pollination, and biotic responses to climate change [3, 12, 10]. These collections represent a significant societal investment in research [27]. Recent funding [30, 6] coupled with improved and affordable imaging has led to a number of collection digitization efforts worldwide (e.g. [17, 4, 38, 37]).

Insect collections typically have the organisms mounted on pins with labels beneath each specimen (Fig. 1), and specimens in boxes with multiple boxes in a drawer. Some have obtained high-resolution images of whole drawers [8, 29, 24], although these views do not afford a clear view of the labels. InvertNet [18] created a robotic camera mount to image drawers and rotation of the camera affords some ability to read the labels. Some have explored the use of "crowd sourcing" to read the labels [13, 15]. Finland's Digitarium is a semi-automated system that includes removal of the labels from the pin for imaging [34, 32], producing a digital image of the label, however, the textual information must still be transcribed.

Prior efforts to capture the labels have dismissed optical character recognition (OCR) as unreliable due to sensitivity to image quality. Labels beneath the specimens contain location, species identification, and other information thus digital capture of the label information as text enables powerful query-based exploration. The focus of this paper is on methods to extract the label text into digital records.

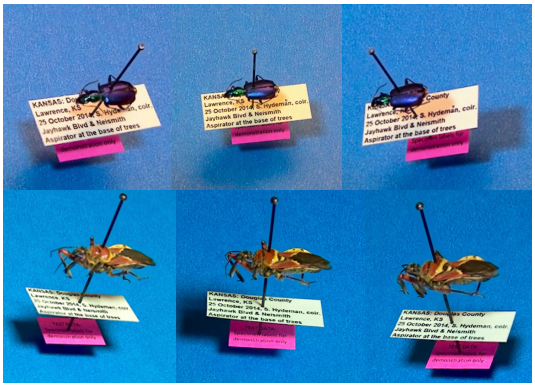

Figure 1. Images of pinned insect specimens acquired from three different camera views. Notice the occlusion of the labels from the insect, other labels and the pin.

Our goal is to develop an automated pipeline for capture of label information that does not require handling or removal of the label from the pin, but captures all the text from the label. With collections containing millions of specimens,the processing should be on the order of 1-2 seconds per pin. Our preliminary experiments indicated that OCR can provide successful transcription of text *if* the input image is high quality. Our solution was to use a multi-camera system to collect multiple views of each label and build a composite image from these occluded fragmentary images that can then be read by OCR. The prototype system uses commercial light-field cameras [28, 23] that have an extended depth of field suitable for imaging at shallow angles that enable imaging of labels stacked on the pin. The cameras allow dynamic re-focusing and produce depth information [33, 7] useful for segmentation of the images to identify the labels.

While other approaches to image and transcribe text on labels have required removal of the label from the pin, we

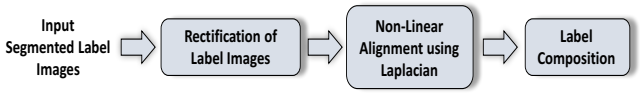

Figure 2. Our pipeline to compute a single rectified-composited image of a label from multiple camera viewpoints.

are not aware of any *automated* solutions for reading the labels without removal from the pin. Our work, however, does have some similarity with other text extraction techniques, especially methods for reading text from images of books (i.e. "scanning" a page of text from an image of an open book), a few of these include [16, 19, 22, 20, 35]. Imaged pages of books have many lines of text, often on curved surfaces [39], and approaches to "flatten" these images derive the surface geometry by identifying the text lines (from which the deformed 2D grid of the text in space can be transformed to a rectilinear grid). These approaches require *many* lines of text to accurately determine the layout of the text lines on the page. The labels on pinned insects contain only a few lines of text (range is usually 1-5 lines, with 3-4 lines of text appearing most frequently in the collections we have been using). With only a few lines of text, there is insufficient information to apply the book-scanning techniques [16, 21, 35] directly. Evaluation of available software from these prior approaches on our label image data provided unsatisfactory results.

The collections contain a diversity of formats (text layout, text spacing on the label, inter-label spacing on the pin, fonts, relative orientation of stacked labels, etc). The insect specimens and multiple labels on a pin occlude the labels. Some labels are bent or deformed. We have developed an approach that handles small, fragmentary label images, extreme viewing angles, and occluded regions (from both the pin and the specimen). Below we present details of our approach and provide experimental demonstration using both museum specimens and experimental test labels.

2. Methods in our Pipeline

Our goal is to obtain an image of the label that is OCR friendly. We use multiple light field (LF) images of a pinned insect specimen and the labels taken from different camera views and build rectified label images that can be composited to form our OCR-ready image. We make two assumptions about the input label images. First, we assume that the contour of the input labels form a rectangle, where the text lines are parallel to the longer edge and orthogonal to the shorter edge of the label in camera space. Second, we assume that there is very little deformation along the short edge of the label. Due to the size of the labels, this is a reasonable assumption because deformation of these labels is typically caused by the effects of gravity. We do not make any assumption on the kind of deformation for longer edge.

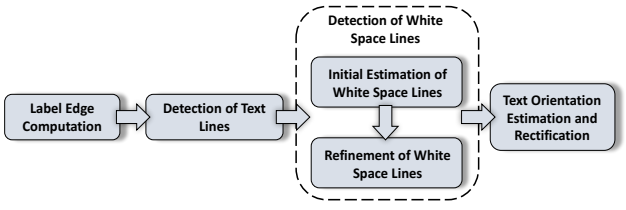

Figure 3. Workflow to rectify a single label image.

The input to our pipeline are multiple, focused LF images of different views of a specimen. We segment multiple labels that may be present in a single LF image and then find their corresponding labels across multiple LF images using their depth information. Due to space limitations, details of this step are omitted here. Given a set of these corresponding segmented label images, we first rectify each label image to get a fronto-parallel view. We then align all these images together using feature based non-linear alignment technique followed by a composition step to generate a single OCR friendly image of the complete label (Fig. 2). We now describe each of these steps in detail.

2.1. Rectification of Label Images

In order to rectify the label images, we build a two-dimensional coordinate grid which faithfully represents the underlying text on the labels, i.e. one coordinate along the text line and the other across the text lines. For this we first compute the four edges of the labels. Because there is no guarantee on the number of text lines present on the label (typically fewer than 5), we use the edges to assist us in finding these coordinate lines. We then compute the location of the text lines and white space lines followed by text orientation estimation. Using these we build the two-dimensional grid and rectify each label image to get a fronto-parallel view as if the camera was directly over it (Fig. 3).

2.1.1 Label Edge Computation

From each segmented label image, we extract the four edges of the label. The contours of labels may have arbitrary shape due to occlusion. To determine label orientation we approximate the bounding box using a modified oriented bounding box (OBB) [1] computation (where we use the convex hull because the edge contours of occluded labels cause unstable behavior of OBB computation).

First, we find the line passing through the two short edges using peaks in Hough Transform near the rotation angle of the OBB and determine the pair of edges from all candidates edges having maximum distance between them (Fig. 4b). Using these edges/lines we then compute the four corners of the label. We do not repeat this process for the longer label lines because the longer edges of the label often have warping (and do not appear as lines). Instead,

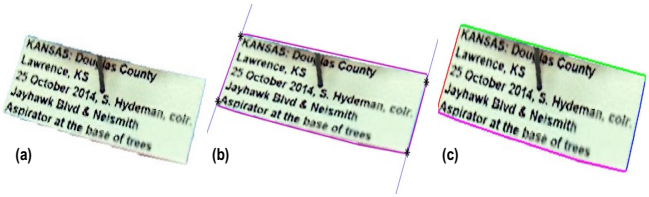

Figure 4. (a) Label images from Fig. 1 lower row. (b) Shorter edges (blue), resampled-convex hull (magenta) and the four corners (black) are used to compute (c) the four edges of the label.

we find the intersection of the convex hull computed earlier with both the shorter label edges. Using the corners of the label and the convex hull, we then find all the four edges of the label by solving for the shortest path between the corner points using Dijkstra's algorithm, where points on the convex hull have zero and rest of the points have a weight of 1 (Fig. 4c).

2.1.2 Detection of Text Lines

In order to compute points which lie on lines of the text, we analyze the intensity profiles between the two long edges of the label. After smoothing, we compute the valleys in the smooth-intensity profiles as the location of points on the text as shown in Figure 5a and 5b. The motivation behind this step is that we want to compute these text points without relying on noise-sensitive operations such as image binarization and character segmentation.

Using these dense set of points which potentially lie on the text of the label, we now cluster them such that each cluster represents a single line of text. This is challenging because the pin may contribute distracting points and occlusion may lead to large gaps within a line of text. To address both these issues, we first compute a vector V_T, orthogonal to both the short edges of the label, as the most likely orientation for the text lines. Using this we then construct a weighted graph G from the set of text points P_T using

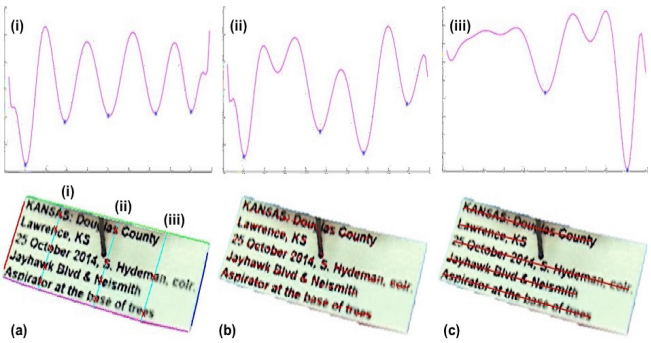

Figure 5. (a) Intensity profiles (top row) of points (cyan) between label edges. Valleys (blue) are possible locations for text (red). Clustering these points (b) finds multiple text lines (c). Notice our algorithm does not cluster points lying on the pin.

Delaunay Triangulation. Given two vertices v_i and v_j in G we compute the edge weight W_{ij} as the product of its length dis_{ij} and the angle which the vector $v_i - v_j$ makes with V_T. The motivation behind choosing such a weighting function is to assign higher weights to edges between two text lines (these are generally longer and not parallel to the text) and lower weights to edges along a single text line. Furthermore, points which lie extremely close to each other will produce unreliable orientation estimates and hence we assign their weight to be 0.

$$W_{ij} = \begin{cases} dis_{ij}.(1 - \Delta v) & dis_{ij} > 1 pixels \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where

$$\Delta v = abs\left(dot\left(\frac{v_i - v_j}{\|v_i - v_j\|}, \frac{V_T}{\|V_T\|}\right)\right)$$

(Note this weighting is based on the cosine of the angle).

Using this weighted graph G we continuously prune edges with high weights in an attempt to remove edges present between text lines, resulting in a graph with disconnected components. At each iteration we check whether the vertices of the disconnected components form potential text lines by fitting a cubic polynomial and computing the fitting error. We stop pruning edges when the ratio of the sum of the fitting errors from all disconnected components of current and the previous iteration does not change, suggesting that we are now pruning edges along the text lines. In order to achieve faster convergence we first perform a minimum spanning tree (MST) on G and then prune edges on this MST to obtain all text lines (Fig. 5c).

2.1.3 Detection of White Space Lines

Using the text lines computed above, we first compute an initial estimation of the white space lines. These are later refined so that each pair of adjacent white space lines sandwich a text-line. To get an initial estimation of the white space lines, we assume that they lie close to the middle of two adjacent text lines. Hence, we first construct a Constrained Delaunay Triangulation (CDT) [11] using points

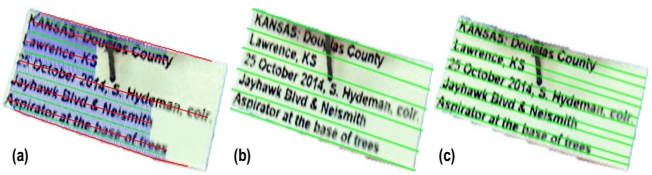

Figure 6. (a) Using the text lines, we construct the CDT (blue) and compute its Voronoi vertices (green) to get an initial estimation of points lying on white space lines. This initial estimation (b) is then refined to obtain pairs of white space lines which sandwich individual text lines (c).

on the text lines as shown in Figure 6a. Voronoi vertices in the dual Voronoi diagram corresponds to points representing the initial white space lines (Fig. 6b).

The initial white space lines are not accurate enough for warping and rectifying these small labels. We refine these white space lines by adjusting the white space line points to sit on top and bottom of the text sandwiching the text lines. For this, we first compute the points corresponding to the top and bottom of the text by finding the peaks in the intensity profile of the points sandwiched by the two adjacent white space lines (we expect a single valley corresponding to the single line of text). From the two sets of white space line points lying above and below a text line $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, we separate those points which do not have any text between $\{(x_1, y_1), (x_2, y_2), ..., (x_s, y_s)\}$ and for the remaining $n - s$ points maximize the following objective function:

$$E(\delta y_1, \delta y_2, ..., \delta y_i) = \sum \phi_i(\delta y_i) + \lambda \sum \psi_{i,i+1}(\delta y_i; \delta y_{i+1}) \quad \forall i \in n - s \quad (2)$$

where δy_i is the vertical shift of points (x_i, y_i), $\phi_i(\delta y_i)$ measures the log-likelihood of a shifted point $(x_i, y_i + \delta y_i)$ being at the true top or bottom of the text and $\psi_{i,i+i}(\delta y_i; \delta y_{i+1})$ is the smoothness measure that penalizes sharp changes in the slope between these points. Note the shift δy_i in the white space line points above and below are in opposite directions. We solve the above objective function using dynamic programming as in [35]. The vertical shift δy_i for remaining points $\{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i) \forall i \in s\}$ is computed from the median of $\{\delta y_i \forall i \in n - s\}$. This results in the computation of accurately refined white space lines as shown in Figure 6c.

2.1.4 Text Orientation Estimation and Rectification

Because there may not be enough text on the label to compute the text orientation and the presence of the pin which may interfere in computing local stroke statistics as in [35], we use both the shorter edges of the label as guide to accurately compute the text orientation across the entire label.

We uniformly sample the text lines computed earlier into K coarse points - points where we want to compute the text orientation. We then select a patch around each of these points such that there is a small overlap between neighboring patches and construct a local histogram of orientations for every patch. For each of these patches we then select the top M orientations having maximum magnitude as candidate text orientations dominant in those patches. We linearly interpolate the orientation of the two shorter edges across the sample points and find the closest to the interpolated orientations among the M candidate text orientations. These then become the final dominant text orientations for those patches. For patches which are not lying on any text, we use the interpolated orientation as their final text orientation. By interpolating the orientations of the two shorter edges of the label, we remove any bias that may be introduced due to the pin and also compute the most plausible text orientation even in those patches lying on the blank regions of the label (Fig. 7a). We empirically found best results for K, M=15.

Having computed both the text orientation and the white space lines accurately, we now can construct the 2D coordinate grid which faithfully represents the underlying text on the label (Fig. 7b). We use the 2D coordinate grid to rectify these label images to get a *fronto-parallel* OCR friendly image (Fig. 7c).

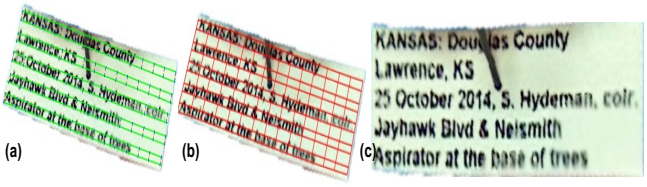

Figure 7. (a) Regions around text lines are used to find a dominant orientation guided by the side edges orientation. (b) A 2D coordinate grid is constructed and then used to rectify the label (c).

2.2. Non-Linear alignment using Laplacian

Using the above rectification pipeline, we rectify all the segmented label images from a pinned insect specimen captured from different camera views. Usually no single image contains the complete label as it has occlusions from other labels and/or the the pinned insect, so we create a single composite image from these rectified labels. For this composite image to be OCR friendly, it is necessary to have an accurate *text-to-text* alignment within multiple label images. Hence we first coarsely align these rectified label images followed by a non-linear alignment such that text from all text lines is accurately aligned among these label images. Given two rectified label images, we now describe the process for this non-linear alignment.

Affine Alignment: We first resolve any scale and translation component between the two label images that may be present after rectification using a global affine transformation to align images pairwise (Fig. 8b). We compute correspondences between images using scale invariant SURF features and descriptors [5] and subsequently remove any outliers using RANSAC. Given these robust correspondences, we solve for affine transformation using linear least square formulation [2] to get coarse alignment (Fig. 8c).

Non-Linear Alignment: After affine alignment we perform a non-linear alignment where we compute dense correspondences between these two affinely aligned label images and use them to solve the Laplace's equations with boundary conditions.

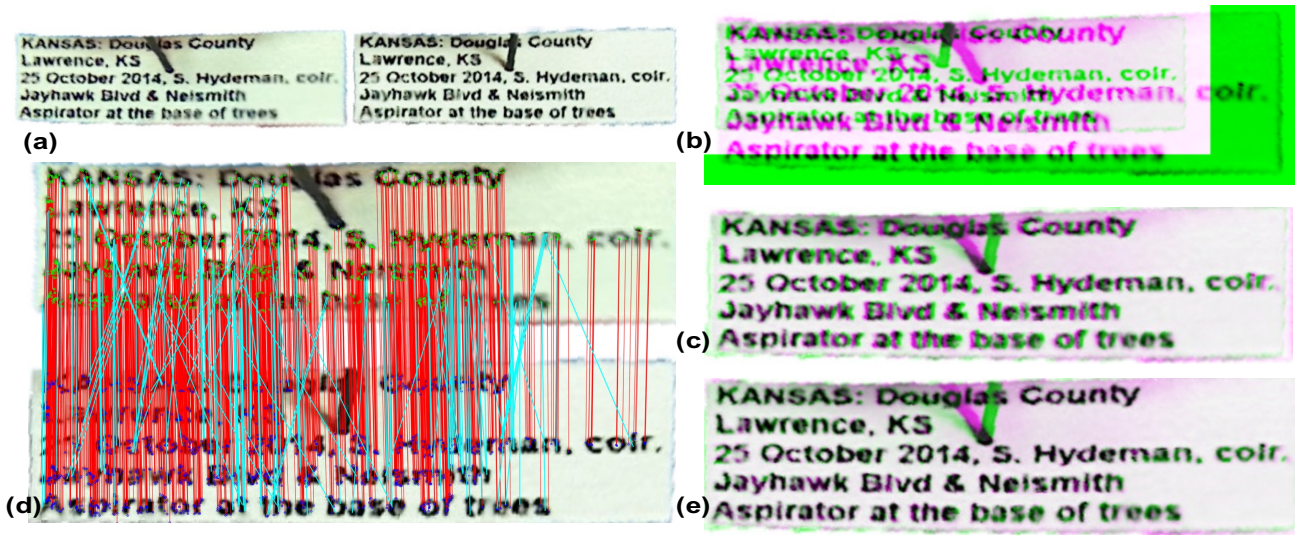

Figure 8. (a) Non-linear alignment of two rectified label images from lower row of Fig 1. (b) Overlay before and (c) after affine alignment. (d) Dense correspondences (red). Note the smooth displacement field of the correspondences in both x and y direction, used for outlier (cyan) detection. (e) Final non-linear alignment of text using Laplace Equations with Dirichlet Boundary conditions.

To ensure dense correspondences between the two label images, we first compute DAISY feature descriptors [36] for every pixel in both images. We are only interested in the alignment of text in the two label images thus we only align regions around each individual text lines. Using image regions containing only one "strip" of text, we first divide it into patches and compute correspondences for only a few keypoints inside each patch (Fig. 9a) as described in Algorithm 1.

To get correspondences which are distributed uniformly across the text strips, we divide each text strip into 20 patches with small overlap such that 10 patches each lie above and below the text line. Given one patch p in text strip T_1 and its neighboring patches q in the other text strip T_2, we first compute feature keypoint locations using Harris [14] and MSER [25] operators in both p and q and then match the DAISY descriptors for these keypoints locations. Let keypoints in patch p be R and in patch q be M. We then compute R_i'', the corresponding point of R_i in M that minimizes the euclidean distance between the DAISY descriptors. To ensure robust correspondence for R_i, we also compute R_i''' by comparing the DAISY descriptors of pixels within a small neighborhood Ω of R_i in T_2. Finally we compute $R\prime_i$, the final correspondence for R_i by taking the minimum between R_i'' and R_i'''. Throughout our experiments we used Ω=10 pixels.

Algorithm 1 Feature matching between the two text strips

Input: Two text strips T_1 & T_2.
Output: Dense correspondences K between them.

Divide T_1 and T_2 into 20 uniform patches.
$K \leftarrow \phi$
for each p patches in T_1 **do**
  $q \leftarrow$ Neighbouring patches in T_2
  $R \leftarrow$ Feature keypoints in p.
  $M \leftarrow$ Feature keypoints in q.
  for each feature points $R_i \in$ R **do**
    $R_i'' \leftarrow$ Eucledian(Daisy(R_i),Daisy(M))
    $R_i''' \leftarrow$ Eucledian(Daisy(R_i),Daisy(Neigbours)
      ▷ Neighbours $\leftarrow$ pixels $\in \Omega$ of R_i in T_2
    $R_i' \leftarrow \min(R_i'', R_i''')$
    add R_i' to K.
return K

After computing the correspondences from each pair of text lines we remove any outliers that may be present by analyzing the displacement in the correspondences in both the x and y direction as shown in Figure 8d (we expect the displacement field in both x and y direction to be smoothly varying). Using these accurate correspondences we solve the Laplace's Equations with Dirichlet boundary conditions which accurately matches points where we have correspondences and smoothly interpolates rest of the points [2]. Using the above alignment technique, we align all the label images of pinned insect specimen to a single label image to get a set of aligned-rectified label images (Fig. 9b).

2.3. Label Composition

As the final step of our pipeline, we perform compositing of these aligned-rectified label images. Although there is a rich literature on image composition using image pyramids [9], gradient based techniques [31], exposure fusion [26], etc, most of these techniques will fail unless we first segment out the pin. We used a naive approach where for

Figure 9. (a) Dense correspondences computed at feature keypoints between corresponding text-strips from two labels. (b) Set of 3 aligned-rectified label images from lower row of Fig. 1.

each pixel in the composited image we compute the max value at that pixel across all the aligned label images to get rectified-composited image as shown in Figure 10. This simple method resulted in images that were OCR-ready. We are currently evaluating the efficacy of using this simple approach versus other compositing algorithms.

Figure 10. Two results of label composition using 3 views (left column are from top row of Fig. 1, right column are results from bottom row). Composition using the maximum value (top row) at each pixel removes the pin from the final composite image. Composition using the minimum value at each pixel (middle row) shows the accuracy of our non-linear text alignment process. For comparison we also performed composition using exposure fusion [26] (bottom row). A gap is seen in the composite image in the first column due to occlusion in all 3 views.

3. Experimental Results

Datasets: We perform evaluation of our method on two types of datasets: an experimental data set that we created and museum specimens. Our real data, comprising of 25 pinned insects specimens from the Chicago Field Museum of Natural History (FMNH), had at least two labels per specimen and displayed a variety of issues mentioned in section 1. Because these were actual specimens, we could not perform any evaluations that involved extracting the labels which might damage the specimens. To permit careful validation of our approach, we created 40 test labels by printing text (replicated from museum labels) onto card stock paper similar to that used in the museum archives and a font and a font size similar to that of actual specimen labels. To test the robustness of our method, the test labels provide a good representation of various scenarios that one may encounter while digitizing the labels, including: text lines of varying lengths, uneven line spacing, different number of text lines (ranging from 1-7), different spacing within a single text line, different indentations of text, both upper and lowercase text, text comprising of both alphabetical characters and symbols (Fig. 11). The test labels were scanned at 600 DPI and then stacked (on average two per specimen) by casually aligning them using a museum specimen pin. The test labels, however, did not have a specimen on the pin (and thus the only occlusion of the top label was from the pin). The labels in both these datasets were imaged using a camera rig setup comprising of three first generation Lyro [23] cameras capable of acquiring light field images in a single snapshot. The cameras were held in a rig that allowed us to simultaneously acquire 3 images. For a "production" pinned insect pipeline, we plan to use more than three cameras, however for the label text acquisition, three cameras was sufficient for algorithmic development and testing.

Evaluation: Due to the fragile nature of the specimens from FMNH, we only provide qualitative results of our method on these labels (Fig. 14). Using the 40 test labels we provide both qualitative (Fig. 11) and quantitative results of our method. We validate our pipeline by computing the maximum value of the normalized cross-correlation (*validation score*) between the rectified-composited image obtained from our method and its scanned image. The scanned label is, essentially, the best case scenario for systems that remove the label from the pin. Comparing our output with the scan, thus provides a metric to the best possible previous methods of removing and scanning labels.

In order to evaluate potential bottlenecks, we measure the accuracy of various components in our pipeline. We check the robustness of white space lines by computing the total maximum error in the text width of the label from different camera views. Figure 12 shows a plot of the maximum text width error with the validation score, where the

Figure 11. Coordinate grid (bottom row) computed using our method on a set of challenging labels (top row). These include labels with single text line, centered text, text on a curved label, and six text lines. Note the robustness of our method to the specimen pin and its shadow on the label. Using these grids we can successfully rectify these label images. Please see supplementary materials for more examples.

Table 1. Ablation study of Non-Linear Registration

	Mean SSD	Validation Score
without Non-Linear	699.89 ± 169.76	0.67 ± 0.07
with Non-Linear	428.71 ± 114.28	0.74 ± 0.04

strong correlation suggests that labels with high text width error could result in bad composite image. We also measure the accuracy of the non-linear registration by computing the sum of squared errors (SSD) between the aligned labels. We normalize it by the number of pixels in the image and take the mean value as a measure of image-consistency after alignment. Figure 13 shows a plot of the mean normalized SSD with the validation score. The strong correlation in the plot suggests that labels with high mean normalized SSD might not result in OCR friendly composite image. Both these scatter plots suggest that errors from one step might propagate to the final composite image. Hence, such error metrics help assess the likelihood of the composite image to have high validation score.

We also studied the effect of non-linear registration in our pipeline. Effectively, we computed the mean normalized SSD and the validation score with and without (only affine) non-linear registration (Table 1). We find that non-linear registration reduces the mean SSD by 38% and increase the validation score by 0.07. To further assess the quality of the composite, we performed a qualitative study where three human subjects visually inspected the final composite image after non-linear and affine registration. Each subject assessed whether the composite image is readable, partially readable or not readable. From the consensus of the three different individuals we found 47.5% (19/40) of the labels to be not human readable without the non-linear registration (Fig. 13).

Though there is ongoing research to improve OCRs, Figure 15 shows some preliminary results of using an off-the-shelf online OCR tool (https://ocr.space) with no post-processing on the composite labels. On our small sample, we achieved just over 80% accuracy with about 15% error and 5% omissions. Using improved OCR tools (such as Google's open source OCR that includes support for

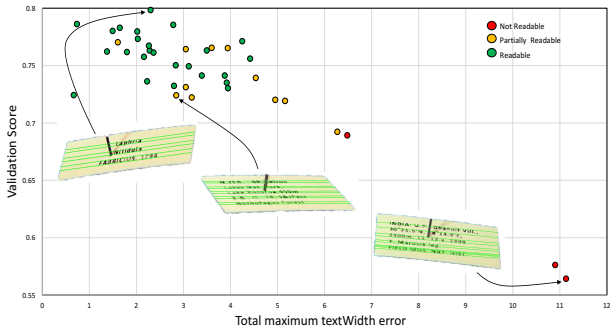

Figure 12. Plot between validation score and total maximum text width error from 40 composite test images after computing the white space lines using our method. A strong correlation between the two is seen suggesting labels with high text width error result in bad composite image. Result of visual inspection (by three different individuals) of the final composite image is shown using colored circles. Note the rightmost labels have high errors in text width resulting in bad composite images (Fig 13).

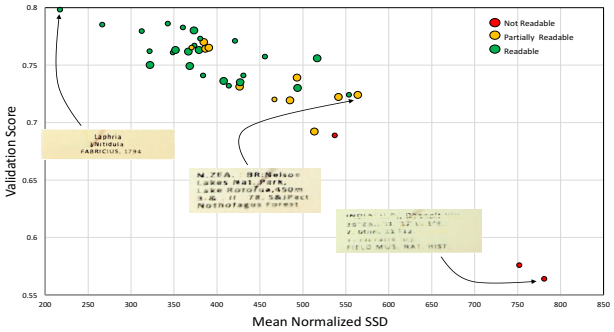

Figure 13. Plot between validation score and mean normalized SSD from 40 composite test images after non-linear registration. A strong correlation between the two is seen. Result of visual inspection (by three different individuals) of the final composite image is shown using colored circles. The increase in size of the circle indicates the composite image, which earlier was not readable (when using only affine registration) became partially or completely readable after non-linear registration.

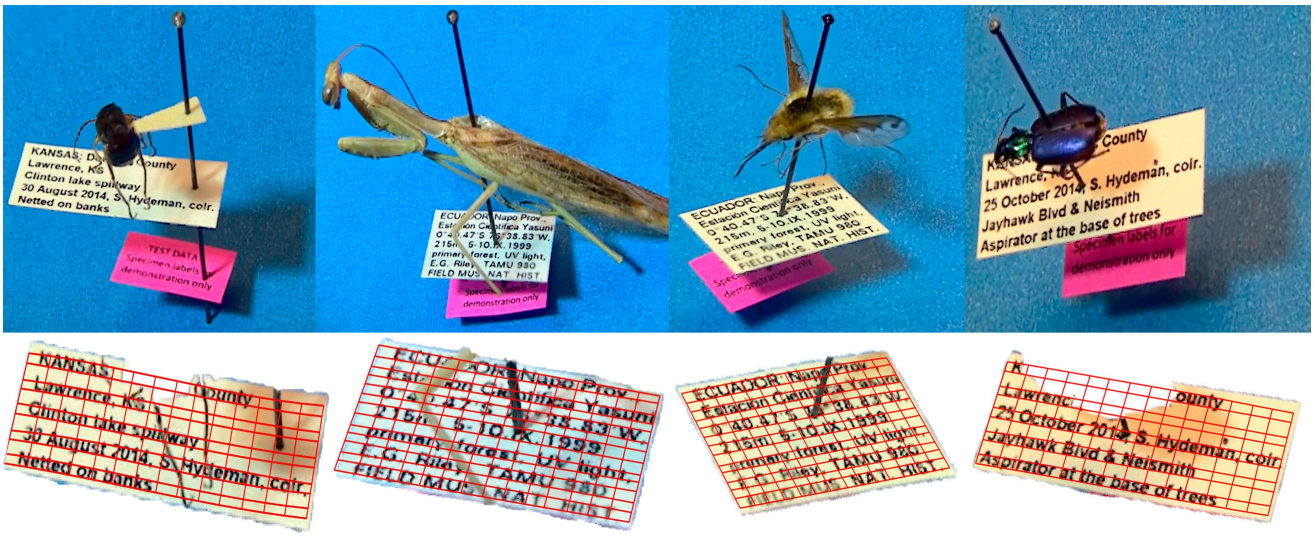

Figure 14. Coordinate grid (bottom row) successfully computed using our method on a set of labels with pinned insects (top row). These labels include occlusions from the insect, labels and the pins and illustrate the acute camera angles necessary to capture the labels due to the different shapes and sizes of the pinned insect. Please see supplementary materials for more examples.

many languages, including Latin) and some post-processing to sharpen the composite image, the OCR results can be greatly improved.

Performance: Our complete label processing pipeline (rectification of a label image, pair-wise non-linear alignment, and final compositing) takes approximately 3 minutes with our unoptimized MATLAB implementation on an Intel Core i5 CPU with 8GB RAM. The most time-consuming step in our pipeline is finding dense correspondences for non-linear alignment which takes up to 2 minutes. We are working on a optimized C++ implementation which should finish the entire pipeline in less than a minute.

4. Discussion

Large collections of pinned insect specimens require a quick capture and processing pipeline to accomplish a goal of digitizing millions of specimens in a 2-3 years time frame. We have developed an image processing pipeline that demonstrates that, with some computation, it is feasible to produce label images that are OCR-ready from images of "intact" pinned insect specimens. The application of OCR will greatly improve the processing time for digital capture of label information. Our prototype system uses light field cameras that can provide an in-focus image for a large depth of field, removing the need for careful focusing after specimen placement. We have demonstrated that even with one or two lines of text, the approach can be used to rectify labels.

There are several avenues for potential future work. First, we have presented results using only 3 cameras. We envision that a "production" pipeline will have views from all sides of the specimen to obtain full coverage of each label. Second, the final compositing step could be improved. For example, after segmenting out the pixels associated with the the pin, a gradient based Poisson compositing could be applied. Third, our approach does not provide consistent results on handwritten text on labels often found in older collections. We have not yet explored methods to handle hand written labels. The approach also exhibits sensitivity to illumination changes. We are currently developing an enclosure that will provide consistent illumination conditions.

Composite Label	OCR result	Test Label	OCR results
Monterey Co.	Monterey Co	Monterey Co.	Monterey Co.
ORE: Grant co., 5000' E of Prairie City Dixie Pass Blue I. vi. 1957. Is Dybas Ber.: garbage & forest sublitter	ORE: Grant co., 5000' E of Prairie City Dixie Pass Blue I. Vt. 1957. Is Dybas floe. d arbage & for • sublitter	ORE: Grant Co., 5000' E of Prairie City Dixie Pass Blue I. vi. 1957. Is Dybas Ber.: garbage & forest sublitter	ORE: Grant co., 5000' E of Prairie City Dixie Pass Blue I. vio 1957. Is Dybas Ber.: garbage & forest sublitter
N. GUINEA :E Highlands Dist. Goroka. El. 5200' IV51971	N. GUINEA itighlands 0ist. GorokD. 5700 IV51971	N. GUINEA :E Highlands Dist. Goroka. El. 5200' IV51971	N. GUINEA Highlands Dist. Goroka. El. 5200' IV51971
Laphria Nitidula FABRICIUS, 1794	Laphr;a •rgjriouio FABRICIUS. t,' 9 4	Laphria Nitidula FABRICIUS, 1794	Laphria Nitidula FABRICIUS, 1794

Figure 15. OCR results for composite and test labels.

5. Acknowledgement

This material is based upon work supported by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under contract DE-AC02-06CH11357.

References

[1] N. Agarwal, X. Xu, and M. Gopi. Automatic detection of histological artifacts in mouse brain slice images. In *Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging*, pages 105–115. Springer, 2016. 2

[2] N. Agarwal, X. Xu, and M. Gopi. Robust registration of mouse brain slices with severe histological artifacts. In *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*, page 10. ACM, 2016. 4, 5

[3] A. H. Ariño. Approaches to estimating the universe of natural history collections data. *Biodiversity Informatics*, 7(2), 2010. 1

[4] Atlas of Living Australia. http://www.ala.org.au/, 2015. 1

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006. 4

[6] Beyond The Box Digitization Competition. http://https://beyondthebox.aibs.org, 2016. 1

[7] T. E. Bishop and P. Favaro. The light field camera: Extended depth of field, aliasing, and superresolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):972–986, 2012. 1

[8] V. Blagoderov, I. Kitching, T. Simonsen, and V. Smith. Report on trial of satscan tray scanner system by smartdrive ltd. *Nature Proceedings*, 2010. 1

[9] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983. 5

[10] A. Chapman. Uses of primary species-occurrence data, verson 1.0. Technical report, Global Biodiversity Information Facility, Copenhagen, 2005. 1

[11] L. P. Chew. Constrained delaunay triangulations. *Algorithmica*, 4(1-4):97–108, 1989. 3

[12] W. Duckworth, H. Genoways, and C. Rose. Preserving natural science collections: chronicle of our environmental heritage. Technical report, National Institute for the Conversation of Cultural Property, 1993. 1

[13] P. Flemons and P. Berents. Image based digitisation of entomology collections: Leveraging volunteers to increase digitization capacity. *ZooKeys*, (209):203, 2012. 1

[14] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Manchester, UK, 1988. 5

[15] A. Hill, R. Guralnick, A. Smith, A. Sallans, R. Gillespie, M. Denslow, J. Gross, Z. Murrell, T. Conyers, P. Oboyski, et al. The notes from nature tool for unlocking biodiversity records from museum records through citizen science. *ZooKeys*, (209):219, 2012. 1

[16] Z. Huang, J. Gu, G. Meng, and C. Pan. Text line extraction of curved document images using hybrid metric. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, pages 251–255. IEEE, 2015. 2

[17] Integrated Digitized Biology project. http://idigbio.org, 2015. 1

[18] InvertNet Project. Invertnet.org, 2015. 1

[19] L. Jagannathan and C. Jawahar. Perspective correction methods for camera based document analysis. In *Proc. First Int. Workshop on Camera-based Document Analysis and Recognition*, pages 148–154, 2005. 2

[20] J. Liang, D. DeMenthon, and D. Doermann. Flattening curved documents in images. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 338–345. IEEE, 2005. 2

[21] J. Liang, D. DeMenthon, and D. Doermann. Geometric rectification of camera-captured document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):591–605, 2008. 2

[22] S. Lu, B. M. Chen, and C. C. Ko. Perspective rectification of document images using fuzzy set and morphological operations. *Image and Vision Computing*, 23(5):541–553, 2005. 2

[23] Lytro Co. http://lytro.com. 1, 6

[24] B. L. Mantle, J. La Salle, and N. Fisher. Whole-drawer imaging for digital management and curation of a large entomological collection. *ZooKeys*, (209):147, 2012. 1

[25] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 5

[26] T. Mertens, J. Kautz, and F. Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. In *Computer Graphics Forum*, volume 28, pages 161–171. Wiley Online Library, 2009. 5, 6

[27] Network Integrated Bio-collections Alliance. http://digbiocol.files.wordpress.com, 2010. 1

[28] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11):1–11, 2005. 1

[29] North Carolina State University Insect Museum. http://insectmuseum.org, 2015. 1

[30] NSF Advancing Digitization of Biological Collections (ADBC) program. http://www.nsf.gov/pubs/2011/nsf11567/nsf11567.htm, 2015. 1

[31] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003. 5

[32] H. Saarenmaa, J. Karppinen, R. Tegelberg, and Z. Wu. World's first automated mass digitization line for pinned insects. In *XXV International Congress of Entomology*, pages 251–255, Orlando, FL, September 2016. 1

[33] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 673–680, 2013. 1

[34] R. Tegelberg, T. Mononen, and H. Saarenmaa. High-performance digitization of natural history collections: Automated imaging lines for herbarium and insect specimens. *Taxon*, 63(6):1307–1313, 2014. 1

[35] Y. Tian and S. G. Narasimhan. Rectification and 3d reconstruction of curved document images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 377–384. IEEE, 2011. 2, 4

[36] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830, 2010. 5

[37] VertNet. http://www.vertnet.org, 2015. 1

[38] ViBRANT. http://vbrant.eu/, 2015. 1

[39] S. You, Y. Matsushita, S. Sinha, Y. Bou, and K. Ikeuchi. Multiview rectification of folded documents. *arXiv preprint arXiv:1606.00166*, 2016. 2