FIGURE DETECTION AND PART LABEL EXTRACTION FROM PATENT DRAWING IMAGES

Jaco Cronje

Council for Scientific and Industrial Research, Pretoria, South Africa
Email: jcronje@csir.co.za

ABSTRACT

The US Patent and Trademark Office, together with the NASA Tournament Lab, launched a contest to develop specialized algorithms to help bring the seven million patents presently in the patent archive into the digital age. The contest was hosted by TopCoder.com, the largest competitive online software developer community. The challenge was to detect, segment and recognize figures, captions and part labels from patent drawing images. The solution presented in this work was the winning submission.

Index Terms— Image analysis, Character recognition, Image segmentation, Document image analysis

1. INTRODUCTION

Around seven million patents are presently stored in the US Patent and Trademark Office (USPTO) patent archive. Many of these patents are originally created before the digital age. Images of the scanned versions of these old dated patents are stored in the patent archive. These documents contain descriptive information as well as drawings about the patent. Most of the drawings are mechanical drawings which contain a lot of parts. Each part is labeled such that it can be referenced from the text description. The figures also contain captions that are used to identify and reference each specific figure.

The USPTO, together with the Harvard-NASA Tournament Lab launched an innovation challenge to invite developers and academics to develop specialized algorithms to detect and label figures and parts from the USPTO patent archive. The evaluation and submission interface to the challenge were hosted by TopCoder.com. TopCoder [1] hosts the world's largest competitive community for software developers and digital creators with a community of over 380,000 members around the world. Up to $50,000 of prizes were distributed to contest winners. The challenge ran for four weeks from mid December 2011 to mid January 2012.

Harvard University concurrently ran a research project about a study on how competitors work together within such contests. All registered competitors were divided into teams of two. The protocol used to match competitors to form teams is described in [2]. Each week during the contest, competitors had to complete a survey about their progress and their teammates progress. The strategic behavior of TopCoder contestants has been analyzed in [3].

Section 2 describes the problem statement. The algorithm evaluation method, implementation restrictions and limitations are described. Related work is reviewed in section 3. The method used by the author to solve the problem is presented in section 4. Section 5 provides some results produced by the proposed method. Finally section 6 concludes the article.

2. PROBLEM STATEMENT

The problem is to extract useful information from patent drawing pages. Each patent drawing page contains one or more figures. There can also be additional data that do not belong to any of the figures. Each figure has a caption and consists of many parts. Each part is labeled with text (typically a number). Some parts may have multiple labels. The task is to extract the location and caption for each figure and to extract the location and text for each part label.

Figure 1 illustrates the useful information of a patent drawing page for the challenge. It contains 3 figures namely $2A$, $2B$ and $2C$. Each figure has 14, 8 and 8 part labels, a total of 30 part labels for the whole drawing page. The figures are indicated by the blue polygons and the part labels by the red polygons.

The input to the algorithm consists of a raw input image and the patent text data if available for the particular patent. Patent text pages contain text that describes the patent and their drawings, the text usually contain references to figures and part labels. The ground truth of a set of 306 patent drawing pages were created for the purpose of evaluating the algorithms. 178 of these drawing pages were provided as a training set. 35 drawing pages were used for preliminary online testing. The remaining 93 drawing pages were used for the final evaluation to determine the prize winning solutions.

The output of an algorithm is evaluated against the ground truth data. The score for each drawing page is determined by the correctness (S_{corr}) and performance score (S_{perf}). The performance score is based on the run-time (T in seconds)

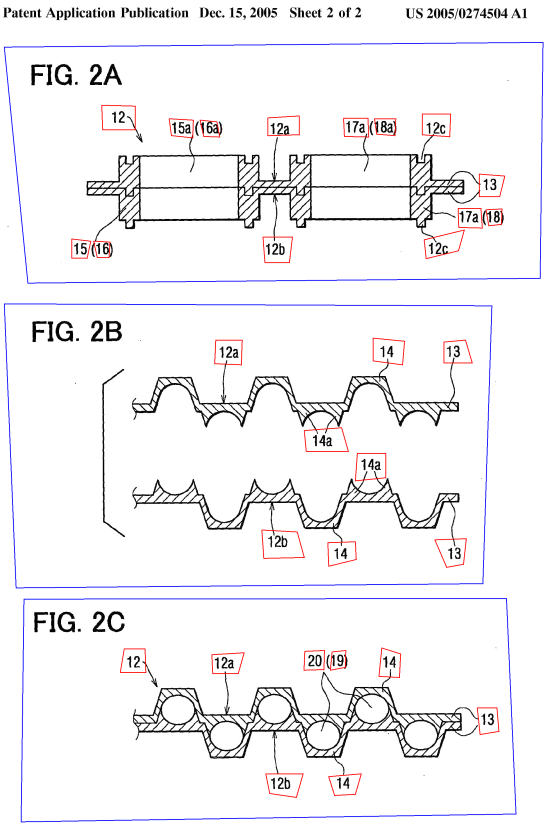

Fig. 1. Example of a patent drawing page with ground truth data. Figures are marked with blue and part labels marked with red polygons.

of the algorithm and calculated by equation 1. No penalty is applied if the run-time is less than a second, but anything slower than that can result up to a 10% penalty.

$$S_{perf} = 0.9 + 0.1 * (\frac{1}{max(T, 1)})^{0.75} \quad (1)$$

The correctness score is calculated by finding the intersection between the bounding boxes of the ground truth data and the algorithms output. For each correctly matched intersection the intersection score is incremented with 0.25 and incremented with another 0.75 if the text for the label or part matches. The intersection score is then used to calculated the precision and recall measurements, which are combined by the harmonic mean 2 to form the final correctness score for the given patent drawing page.

$$S_{corr} = \frac{2 * precision * recall}{precision + recall} \quad (2)$$

The score for an individual test case is given by 3. The overall score is then the sum of scores over all the individual test cases.

$$Score = 1000000 * S_{corr} * S_{perf} \quad (3)$$

Competitors were allowed to program in C++, C#, Visual Basic, Java or Python. The source code size limit was set to 1 MB. No access to external files were allowed. The time limit for each test case was 1 minute and the memory limit 1024 MB.

3. RELATED WORK

An overview of the benefits, requirements and challenges involved in the development of a patent image retrieval framework is provided in [4]. Furthermore, a patent search engine called PatMedia was developed based on the proposed framework. The framework segments the patent drawings into figures, extract their captions and perform feature extraction on each detected figure. The extracted figure features are used to index patent drawings and to search for similar drawings within the patent database. Information extracted from the associated patent text pages are merged with the image based information to improve the performance and resolve ambiguities.

The PATSEEK [5] application is a content-based image retrieval search engine for the US patent database. Just like PatMedia [4], PATSEEK [5] detects the figures from patent drawings and extracts a feature vector for each figure to be used for retrieval purposes. Both of them use slightly different techniques. PATSEEK do not make use of the information in the patent text pages and is outperformed by PatMedia.

The work presented in [6] focus on the extraction of features from patent or technical drawings for retrieval purposes. Lines and their attributes are detected from the drawings. The set of lines is transformed into a nearest neighbor graph and the graph attributes are converted into a 2-Dimensional histogram for fast image comparisons.

The use of angular and radial distribution information for figure feature description was used in [7]. The work in [7] focused thus more on 2-Dimensional shape features in patent drawings.

A method to detect alphanumeric labels from figures is described in [8]. The work doesn't focus specifically on patent drawings, but focuses on documents that contain a mixture of text and figures.

Captions and part labels are extracted from patents in [9] to create a user friendly browser interface. Their approach used an unsupervised clustering algorithm to classify connected components as characters or not. It is assumed that the font used across multiple drawings of the same patent remains the same. The same authors presented a patent drawing segmentation algorithm in [10]. The segmentation algorithm performs Delaunay triangulation to segment the drawing into

a graph. The graph is then further reduced and segmented such that document layout constraints are not violated.

The method presented in this article use similar techniques used in [4], [8] and [9] to extract figure captions and part labels. PatMedia used a commercial Optical Character Recognition (OCR) library where as it was not allowed for the USPTO challenge.

4. METHOD

The method presented in this work was the top submission for the USPTO innovation challenge. Patent drawings usually consist of a header, a typical header can be seen at the top of the drawing page in figure 1. The figures on the drawing may be orientated horizontally or vertically. Section 4.1 describes how the page orientation is detected.

Firstly a margin around the border of the image is cleared to eliminate the header from further image processing steps. The gray scale image is then converted to a binary image by applying a fixed threshold.

Many old patent images contain a lot of salt and pepper noise. A connected component algorithm is performed and if the number of very small components detected are more than 30% of the total number of components, a dilate and erode process are performed to reduce the noise.

4.1. Page orientation

In order to recognize the text from captions and part labels, the orientation of the page needs to be detected. All the connected components that could possibly be a character are used to determine the page orientation. Figure 2 illustrates a patent drawing which is vertically orientated along with its detected connected components.

A voting system classifies the page to be horizontal or vertical. For each character, a vote is cast for a horizontal layout if the width of the character is greater than the height, otherwise a vertical vote is counted. Also, for each character the nearest neighboring character is found. A vote is then cast depending on whether the two characters are more horizontally or vertically aligned to each other.

The dominant orientation with the most votes wins.

4.2. Text extraction

The image is segmented through connected component labeling. Each connected component can be a character, part of a figure or image noise. Each connected component needs to be classified into one of the categories before the figure extraction and part labeling process can proceed.

Components with a width and height smaller than 13 or greater than 250 pixels are regarded as not characters. The resolution of the images were typically 2560 by 3300 pixels. The remaining components are marked as possible characters

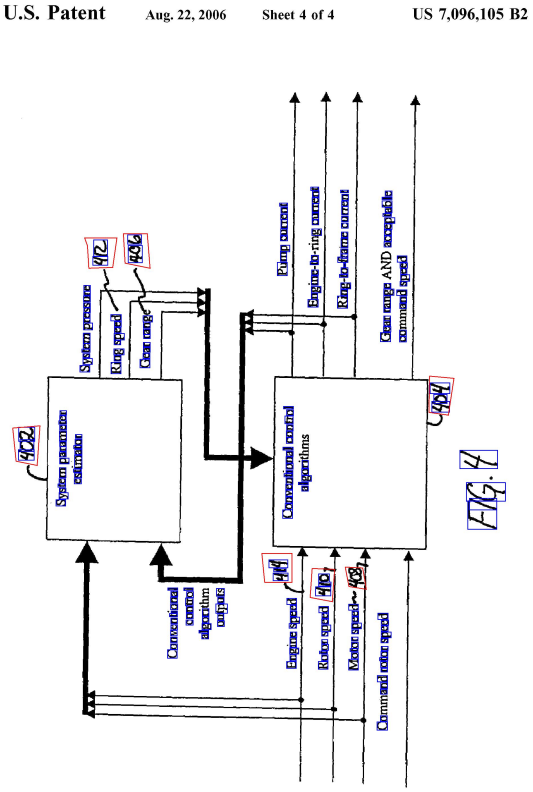

Fig. 2. Vertical page orientation. The connected components that could be characters are indicated with blue rectangles.

if they do not contain any other component within the characters axis aligned bounding box.

Components marked as characters are then sorted from left to right. Groups of character components are created based on the same merging metric described in [8]. The metric merges two components if their horizontal spacing is small and they overlap significantly in the vertical direction.

The group of characters are then recognized. Each character is separately processed by the character recognition system explained in section 4.3.

4.3. Character recognition

A simplistic template matching algorithm is used to perform optical character recognition. Patches containing known characters were manually extracted from the set of training images. Only the ten numerical characters and the characters f, g, a and c were used as templates. The characters f and g had to be recognized to detect the figure captions. The characters a and c mostly appear at the end of part labels and within figure captions. The character b was not recognized because of

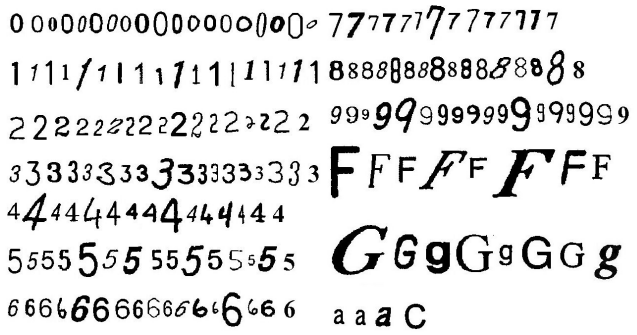

Fig. 3. Characters manually extracted from the set of training images.

the similarities between b and 6. Figure 3 shows the template patches.

The connected component under recognition is firstly scaled to fit an area of 16 by 32 pixels. All the pixels that belong to a hole in the character are marked by using a flood fill algorithm. The scaled image and hole information are compressed into 32 unsigned integers to form the component descriptor.

To find the best matching character, each template is compared with the input component descriptor. The number of matching pixels P and mismatched pixels F are counted. A matching score is calculated by $(P - F)/(P + F)$ and the best scoring template is used as the recognized character.

4.4. Figure extraction

The bounding box and caption of each figure within the drawing need to be extracted and recognized. Firstly the components are extracted as described in section 4.2. Text components that contain the pattern $f1g$ are removed from the component list and added in a list of possible detected figure captions.

A different method is used to segment the figures when no figure caption was detected. Components with an area less than 300^2 pixels are merged with their nearest neighboring component. Larger components are merged only with their intersecting components. Merging two components mean that their axis aligned bounding boxes are merged into one bounding box that contains both of the original bounding boxes. The merging process continues until no more components are merged. Figure 4 shows the components before the merging process.

Each component is initially assigned to their nearest figure caption if captions were detected. Figure 5 shows the components after the merging process. Note that the three components below figure 2B should all be assigned to figure 2B and a simple nearest neighbor assignment will not work in this case and needs to be refined. A segmentation score is calculated by taking into account the bounding box intersecting

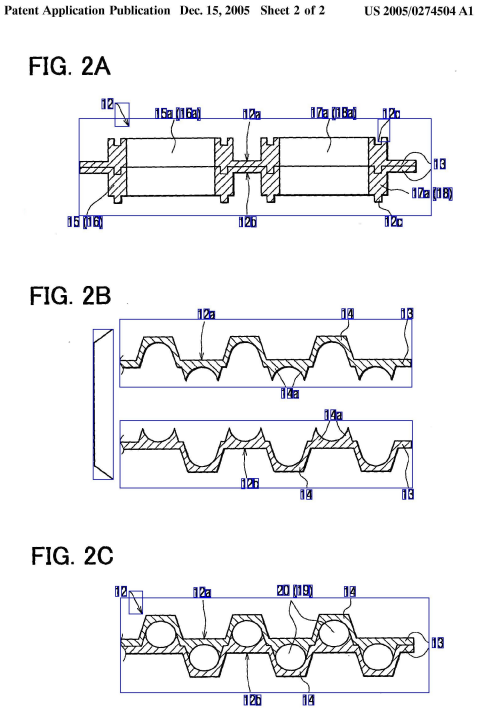

Fig. 4. Components before the merging process begins.

area of the segmented figures. The score is penalized when none or more than one figure caption intersects the bounding box assigned to a figure. The components assignment are randomly shuffled for 1000 iterations and the best scoring segmentation is used.

The header of a patent usually contains text that indicates the current sheet number and the total number of sheets. These sheet numbers are extracted and used to refine the recognized figure captions.

Possible figure captions are extracted from the patent text data and sorted numerically. The recognized figure captions are matched with the captions from the text. The best matching sequence is used for the figure captions in the drawing, taking into account the sheet numbers. For example the last sheet should contain the last figures.

The bounding boxes returned in the output are shrunk such that they minimize their intersection with each other.

4.5. Part labeling

The part labeling process firstly extracts text components described in section 4.2. Patent drawings can contain tables or graphs, usually they do not contain any part label inside their boundaries. The border of each component is examined. If the border is more than 25% filled, the component is considered to be a table or a graph and all the intersecting text

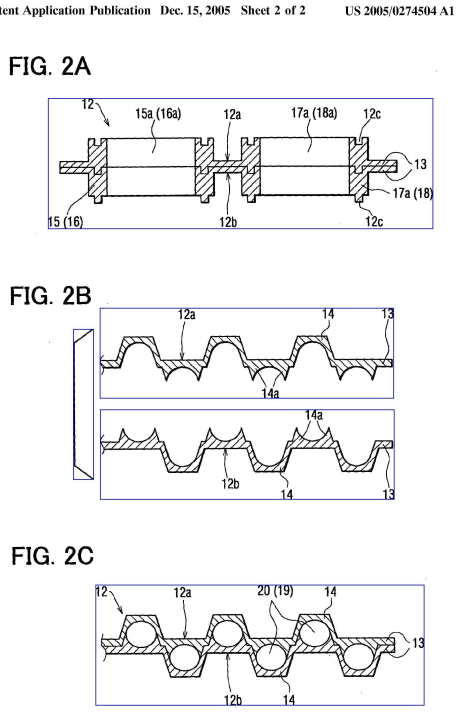

Fig. 5. Components after the merging process.

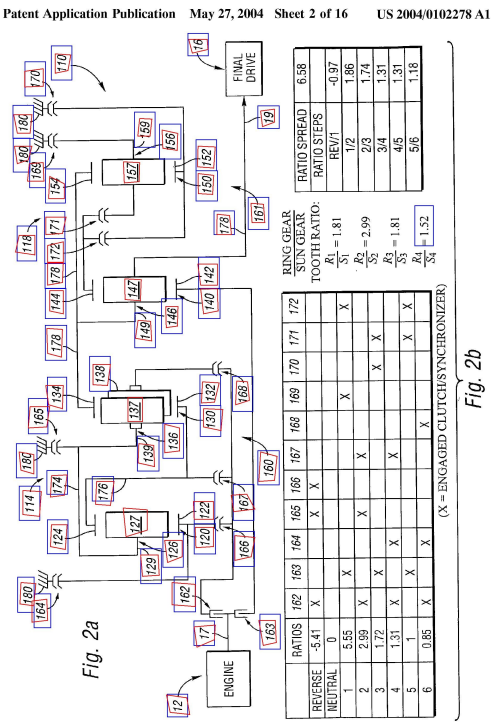

Fig. 6. Patent drawing that contains a table. Red rectangles show the ground truth data and blue rectangles show the detected part labels by the algorithm.

components are removed. Figure 6 shows a patent drawing that contains a table.

Text components containing one of the following characteristics do not classify as part labels:

- The width or height is smaller than 10 pixels.
- The component contains more that 4 characters.
- Figure captions are removed.
- Character recognition matching score below zero.
- No numbers occur within the text.
- The text contains more than one alphabetic character.
- The border surrounding the text is more than 4% filled.

Words that contain numbers are extracted from the patent text data. The recognized text from the remaining text components are corrected by finding the best matched word from the patent text data. The correction only takes place if the character recognition matching score is below 0.5. The text component is removed if the best match from the patent text changed more than half of the original recognized text.

Finally the average height and area of the remaining text components are computed. Any text component where the height or area of which differs significantly from the average is removed from the output. The bounding boxes of the parts are shrunk such that they minimize their intersection with each other.

Table 1. *Training set performance.*

	Correct	Total	Percentage
Figures detected	234	285	82.1
Captions recognized	213	234	91.0
Part labels detected	2875	3752	76.6
Labels recognized	2424	2875	84.3

5. RESULTS

Table 1 shows the performance on the training set. The percentage of correctly segmented figures, recognized captions, part label locations detected and part label text recognized are shown. The running time of the algorithm was below 1 second for all cases, thus avoiding any time penalty. The average recall and precision measurements on the training set is shown in Table 2.

The overall score was 275 million out of a possible 356 million based on the USPTO challenge scoring metric on the training set.

Table 2. *Recall and precision measurements.*

	Recall	Precision
Figures	0.8534	0.8537
Part labels	0.7533	0.7358

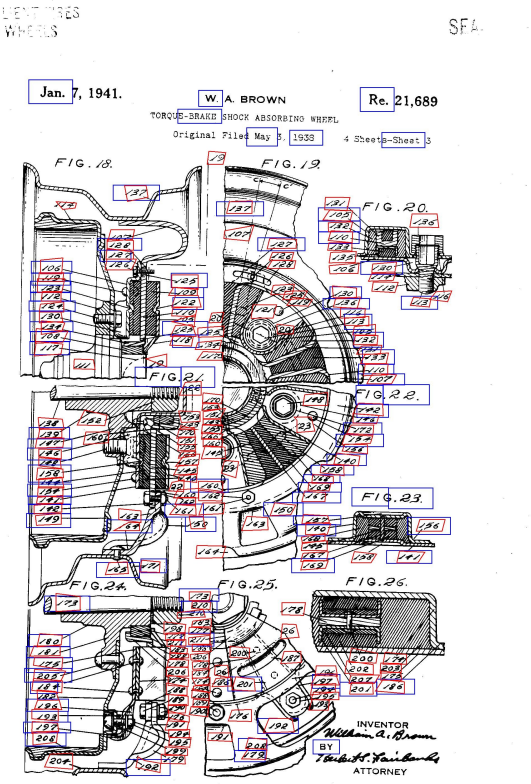

Fig. 7. Patent drawing that contains hand written characters and figures that are difficult to label.

6. CONCLUSION

The work presented in this paper provides a way to segment and label figures from patent drawing pages. A method for part label extraction has been described. The algorithm was tested on a set of real patent drawings and the results look promising as the algorithm scored at the top within the challenge.

There is still room for improvements to the algorithm due to the limited duration of the USPTO innovation challenge. A more sophisticated character recognizer could be integrated. Figure 7 shows a drawing with hand written characters and figures that are difficult to segment.

The USPTO challenge[1] was an interesting challenge and drawn the attention of many top problem solvers around the world. Hopefully more challenges will be launched in the future to promote and encourage academics and developers to solve real world problems together on a global scale.

[1] http://community.topcoder.com/longcontest/stats/?module=ViewOverview&rd=15027

7. REFERENCES

[1] KR Lakhani, D. Garvin, and E. Lonstein, "Topcoder (a): Developing software through crowdsourcing," *HBS Case*, pp. 610–032, 2010.

[2] J.J.M. Tan, "A necessary and sufficient condition for the existence of a complete stable matching," *Journal of Algorithms*, vol. 12, no. 1, pp. 154–178, 1991.

[3] N. Archak, "Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder.com," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 21–30.

[4] S. Vrochidis, S. Papadopoulos, A. Moumtzidou, P. Sidiropoulos, E. Pianta, and I. Kompatsiaris, "Towards content-based patent image retrieval: A framework perspective," *World Patent Information*, vol. 32, no. 2, pp. 94–106, 2010.

[5] A. Tiwari and V. Bansal, "PATSEEK: content based image retrieval system for patent database," in *Proceedings of International Conference on Electronic Business, Beijing, China*, 2004, pp. 1167–1171.

[6] B. Huet, N.J. Kern, G. Guarascio, and B. Merialdo, "Relational skeletons for retrieval in patent drawings," in *Image Processing, 2001. Proceedings. 2001 International Conference on*. IEEE, 2001, vol. 2, pp. 737–740.

[7] Z. Zhiyuan, Z. Juan, and X. Bin, "An outward-appearance patent-image retrieval approach based on the contour-description matrix," in *Frontier of Computer Science and Technology, 2007. FCST 2007. Japan-China Joint Workshop on*. IEEE, 2007, pp. 86–89.

[8] M. Worring and A.W.M. Smeulders, "Content based hypertext creation in text/figure databases," *Series on software engineering and knowledge engineering*, vol. 8, pp. 87–96, 1997.

[9] L. Li and C.L. Tam, "A graphics image processing system," in *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*. IEEE, 2008, pp. 455–462.

[10] L. Li and C.L. Tan, "Associating figures with descriptions for patent documents," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, 2010, pp. 385–392.