Combining Image and Text Features: A Hybrid Approach to Mobile Book Spine Recognition

Sam S. Tsai[1], David Chen[1], Huizhong Chen[1], Cheng-Hsin Hsu[2], Kyu-Han Kim[3], Jatinder P. Singh[4], Bernd Girod[1]
[1]Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA
[2]Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
[3]HP Laboratories, Palo Alto, CA 94304, USA
[4]Deutsche Telekom R&D Laboratories USA, Los Altos, CA 94022, USA
[1]{sstsai, dmchen, hchen2, bgirod}@stanford.edu, [2]chsu@cs.nthu.edu.tw, [3]kyu-han.kim@hp.com, [4]j.singh@telekom.com

ABSTRACT

Despite the successful use of local image features for large-scale object recognition, they are not effective in recognizing book spines on bookshelves. This is because some book spines contain only text components that do not yield distinguishing image features. To overcome this issue, we develop a new approach that combines a text-based spine recognition pipeline with an image feature-based spine recognition pipeline. The text within the book spine image is recognized and used as keywords to search a book spine text database. The image features of the book spine image are searched through a book spine image database. The search results of the two approaches are then carefully combined to form the final result. We implement the proposed hybrid book recognition pipeline used in a book inventory management system, and conduct extensive experiments to evaluate its performance. The experimental results show that while text-based or image feature-based systems only achieve a recall of ∼72%, the proposed hybrid system achieves a recall of ∼91%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval —*Search Process*

General Terms

Algorithm, Design

1. INTRODUCTION

Manually managing items on a bookshelf is a tedious and time consuming task. To solve this issue, research groups have developed automated management systems to identify books on the shelves. One way to recognize the books is to tag each book with an identifier such as an RFID or barcode and read the tag using a specialized reader. Another way is to use images from a digital camera for identifying books [2, 3, 5, 7–9, 12]. Deploying camera-based book recognition solutions is more cost-effective because there is no need to attach physical tags to individual books.

Among the camera-based systems, Chen et al. [2, 3] and Matsushita et al. [9] use image features for robust recognition. In [2, 3], Chen et al. describe a system that builds book inventories using smartphones. They take a picture of a bookshelf and identify the book spines in the image. Furthermore, they use location sensors to attach location information to each book. In [9], Matsushita et al. propose an interactive bookshelf based on a camera-projector system. They use the imagery from the camera to recognize items removed from the bookshelf and project light signals to guide the user to certain items. These systems leverage local image features for recognition and gain invariance to scale change, illumination change, occlusion, and rotation. However, experimentally, we found that their recognition performance was fairly low: a recall of merely ∼72%. Their poor performance on some book spines can be attributed to the fact that image features were developed primarily for natural scenes and thus do not work well with images that have plain texts.

To overcome this issue, we develop a new hybrid recognition system that combines a text-based recognition pipeline and an image feature-based recognition pipeline for more accurate book spine recognition. We use camera-phones to take pictures of a bookshelf. The query image is sent over a network to a server where the bookshelf image is processed. The individual book spines are extracted and passed to the text-based and image feature-based recognition pipelines for recognition. Independently, the two pipelines propose recognition results, and we combined them to form the final recognition result. Our experimental results show that this hybrid system achieves a performance substantially better than those of the text-based and image feature-based systems.

The main contributions of our work are as follows:

- Design and implementation of a novel hybrid book spine recognition system which achieves superior recognition accuracy compared to a text or image feature-based recognition system.
- Development of a method to robustly recognize spine texts from bookshelf images for spine recognition: We extract book spines from the bookshelf image and remove the perspective distortion. Text is localized using a detection method based on Maximally Stable Extremal Regions (MSER) and Stroke Width Transform (SWT) [4] and recognized using Optical Character Recognition (OCR). Recognized texts are used as keywords to search a spine text database with a dictionary built from the spine text in the database.

The rest of this paper is organized as follows. We discuss the related work in Sec. 2. We describe the proposed system in Sec. 3. We report experimental evaluation results in Sec. 4.

2. RELATED WORK

Recognizing books on shelves has been considered [2, 3, 5, 8, 9, 12]. Crasto et al. [5] present an interactive bookshelf based on a camera-projector system where they use the color histogram of the book spines to identify books. Matsushita et al. [9] also introduce an interactive bookshelf while using a different approach for book

The authors would like to thank Area Chair Marcel Worring for all the help during the review process. This work was done while C. Hsu, K. Kim were with Deutsche Telekom R&D Laboratories, USA.

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

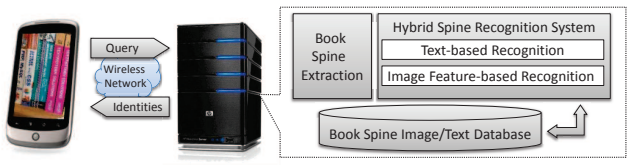

Figure 1: Mobile book spine recognition system.

recognition. Instead of recognizing the book spines directly, they recognize book covers using local image features when books are being removed from the shelf. Loechtefeld et al. [8] suggest using optical tracking methods to identify books from shelves using camera phones while using a pico projector to display guidance information. Chen et al. [2, 3] leverage SURF features [1] for recognizing books using their spine images. They use smartphones to take images of bookshelves and extract the book spines from the images. Each book spine is then queried against a book spine image database. Location information from the mobile device is used to build a location-aware book inventory. In [12], Quoc and Choi develop a framework for recognizing books on bookshelves using robots with cameras. They find individual book spines by detecting the straight lines within the image. From the segmented book spines, they detect the text positions using edges and use character recognition to read the text.

3. MOBILE BOOK SPINE RECOGNITION

The architecture of our system is illustrated in Fig. 1. On the camera-phone, a lightweight application guides the user to take a picture of the bookshelf. The query image of the bookshelf and location data is sent to a server. On the server, book spines are first extracted from the image of the bookshelf. Then, each book spine image is identified using a recognition system which consists of a text-based recognition pipeline and an image feature-based recognition pipeline. The recognized results are sent back to the user and also passed to a book management system that creates a location aware inventory [3] which keeps track of each book's location.

3.1 Book Spine Extraction

We illustrate the steps to extract the individual book spines in Fig. 2. To extract the individual book spines from the image of the bookshelf we first detect lines that resemble book boundaries as shown in Fig. 3(b). Then, we form a quadrilateral that encloses the book spine using the two ends of the boundary lines as vertices. The region within the quadrilateral is extracted from the image and reprojected to form a rectangular region as shown in Fig. 3(c). In the case where a book spine is not fully found due to cropping, we extend the boundary lines to have the same length as the boundary lines of spines that were correctly detected.

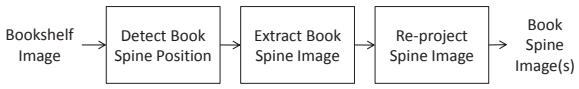

Figure 2: The building blocks to extract book spine images.

3.2 Book Spine Recognition

To recognize the book spines, we match each extracted spine to an online database of book spines. As shown in Fig. 1, we use hybrid recognition system that consists of a text-based recognition pipeline and an image feature-based recognition pipeline. From each book spine image, we detect and recognize the text on the spines and use it as keywords to search a book spine text database. Similarly, image features are extracted from the book spine images and matched to a book spine image database. We combined the results of the two pipelines to form the final output.

Figure 3: (a) Original image of a bookshelf. (b) A line detection algorithm is used to find the spine boundaries from the original image. (c) The extracted book spines are reprojected into rectangle regions.

3.2.1 *Recognizing Spines with Text*

The text on the book spines typically contains the title and the author names, which can provide effective keywords to search for the book. To use the text on the book spines, the text within the extracted book spine image has to be automatically recognized. The process of recognizing the text on the book spines and using the text to identify the book is shown in Fig. 4.

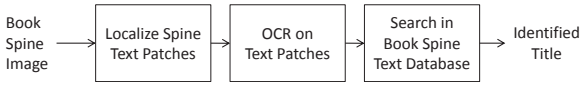

Figure 4: The building blocks for spine recognition based on text.

First, we detect text in the extracted book spine image using a text detection algorithm based on MSER and SWT [4]. MSERs are detected from the image and pruned using Canny edges, forming the character candidates. Stroke widths of the character candidates are found based on distance transforms. Then, they are pairwise linked together based on their geometric property to form text lines. The algorithm localizes the text within the book spine image and also filters out graphical components on the book spine. The localized text is then extracted from the book spine image and denoised using an edge-preserving filter. Fig. 5(a) shows the text patches extracted from the book spine image. Finally, the individual text patches are passed to an OCR engine for recognition. Words that are recognized from each local text patch, as shown in Fig. 5(b), are used as keywords to search a book spine text database.

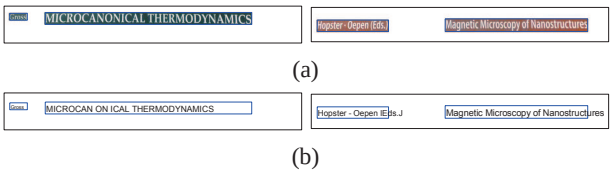

Figure 5: (a) Localized text patches extracted from the book spines. (b) The words recognized using an OCR engine.

To facilitate the search, we organize the book spine text database using inverted files [14] as commonly used in text retrieval systems. First, we construct a dictionary W using the text on the spines of the book title database. For each word $w_i \in W$, we form an inverted file $L(w_i)$ that stores the indexes to the book titles which contain the word w_i. From a query book spine image, we read a set of query keywords Q. We use the keywords $q_j \in Q$ to search through the database. For each q_j, we find a matching dictionary word m_j. We investigate two approaches to find matching words: (1) *Exact word matching*, where we find m_j as the exact same word as q_j. If no exact matching word is found for q_j, then q_j is ignored from Q. (2) *Nearest neighbor word matching*, where we find m_j as the closest distance word to q_j according to the editing distance $d(q_j, m_j)$.

It satisfies the criteria that $d(q_j, m_j) \leq d(q_j, w_i), \forall w_i \in W$. Finally, we calculate the score for the kth book spine as follows:

$$s_t(k) = \sum_j I(k \in L(m_j)), \quad (1)$$

where $I(\cdot)$ is the indicator function which is assigned one if k is in $L(m_j)$ or zero otherwise. We also consider the score when it is further weighted using *tf-idf* (term frequency-inverse document frequency) [13]. *tf* weights the word according to the number of occurrences within the spine text, and *idf* weights the score based on the how many different titles the word has occurred in.

3.2.2 Recognizing Spines with Image Features

In the image feature-based recognition pipeline, we use image features to match the query book spine to a database of book spine images. From the query spine, we extract SURF features [1] and use them to match the spines to a database of book spine images using a vocabulary tree with soft binning [10, 11]. A small set of top scoring candidates from the vocabulary tree are geometrically verified by estimating an affine model between the two spine images using RANSAC [6]. We use the number of consistent feature matches after geometric verification as the score $s_i(k)$. The score of the book spines where no consistent geometrical model is found are set to zero.

3.2.3 A Hybrid Approach

We combine the results of the text-based recognition pipeline with the image feature-based recognition pipeline to form the final result. A linear combination is used, as suggested by [15]. For the text-based recognition pipeline, the score $s_t(k)$ for database spine k is calculated as using Eq. (1). For the image feature-based recognition pipeline, the score $s_i(k)$ for database spine k is the number of feature matches after geometric verification. The hybrid score $s_h(k)$ for book spine k, is calculated by linearly combining scores from the two pipelines as follows:

$$s_h(k) = s_t(k) + \lambda \cdot s_i(k). \quad (2)$$

λ is experimentally determined to be 10. The value roughly corresponds to the ratio of the number of image features to the number of words of typical book spines. Finally, a threshold ϕ is used to determine whether a book spine image is confidently matched. When $s_h(k) > \phi$, the system declares that the book spine is an identified match. Otherwise, the system responds to the user that no match has been found.

4. EXPERIMENTAL RESULTS

To evaluate the recognition performance of our system, we construct a database of 2300 book spine images, which are collected from a library and a book store in Stanford University. We have implemented three book spine recognition algorithms: image feature-based, text-based, and hybrid. We segment book spine images and label the title and authors for each book spine. For the text-based recognition pipeline, we construct a book spine text database from all the labeled text. The resulting dictionary consists of a total of 5398 words. For the image feature-based recognition pipeline, we construct a vocabulary tree using SURF features extracted from the database book spine images. For query images, we took 50 photos of bookshelves using camera-phones, including Nokia N95, N97, and Motorola Droid, with 5MP cameras. We take pictures of bookshelves in different orientations, illuminations, and perspectives.[1]

We use SVGA (1024x768) images, resized from the full-sized images, to evaluate the recognition performance of our hybrid recognition system. Text is detected from each book spine image and the text within the localized patch is recognized using the Tesseract OCR engine.[2] We limit the allowed recognized characters to only alphanumerical letters and a reduced set of punctuations and notations. If not specified, the recognized query keywords are matched to the dictionary using Nearest Neighbor (NN) word matching, and the scores are weighted using *tf-idf*. Image features are extracted from the book spine images and quantized through the vocabulary tree. The top 50 scoring titles are geometrically verified using RANSAC to find the total number of feature matches.

We evaluate the recognition performance using precision and recall on the extracted 454 book spines. The precision is the percentage of correctly identified titles out of the declared correct titles. The recall is the percentage of correctly identified titles out of all query spines. Precision and recall can be varied by adjusting the threshold ϕ, which we use to decide whether the top scoring match is a declared match (Sec. 3.2.3). A lower ϕ means the top scoring match is easily accepted as the correct one, which would lead to lower precision but high recall. A higher ϕ would require a high score for a correct match and leads to higher precision but lower recall.

4.1 Recognition Performance

In Fig. 6, we show the precision-recall curves of the text-based, image feature-based, and hybrid system. Compared to the image feature-based system, text-based system achieves higher recall when the precision is lower than 87%. The image feature-based system achieves higher recall when the precision is higher. At a precision of 95%, the text-based system achieves a recall of less than 40% while the image feature-based system achieves a recall of 72%. Both systems are substantially outperformed by the proposed hybrid system. At a precision of 95%, the proposed system has a recall of 91%, which is 51% and 19% higher that the text-based and image feature-based systems, respectively.

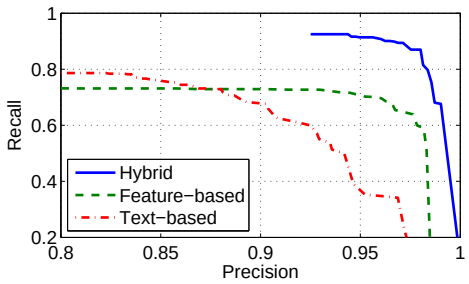

Figure 6: The recognition performance of the three systems.

To gain insights on why the proposed hybrid system substantially outperforms the other two systems, we closely examine the book spines that are correctly recognized by text-based and image feature-based systems. The number of spines for which both identify correctly is ∼58% of total number of spines. This indicates that the two systems are suitable to different types of spines. As shown in Fig. 7(a), spines with text that have generic fonts tend to be harder for the image feature-based system to recognize due to the similarity between visual features. However, character recognition on generic fonts has higher accuracy because the OCR engine is trained to recognize these fonts. On the other hand, spines with graphical components and cursive text, such as the spine shown in Fig. 7(b), are rather challenging to OCR engines. In contrast, image features of these spines are fairly distinctive. Hence, by combining the text-based and image feature-based systems, we mitigate the misses and improve the overall recognition performance.

We investigate how the system performs when combined using a different λ. In Fig. 8, we show how λ affects the recall at different precisions. The best recall is obtained at a λ value of ∼10. We observe that the best λ is roughly the same at various precisions.

We evaluate the recognition latency on a 3.2 GHz i7 server with 8 cores and 6 GB memory. Spine extraction is performed in 0.10

[1] http://msw3.stanford.edu/∼sstsai/BookSpineSearch

[2] http://code.google.com/p/tesseract-ocr/

(a) (b)

Figure 7: (a) Spines with generic text can be recognized by the text-based system but not the image feature-based system. (b) Spines that vary more in style can be recognized by the image feature-based system but not the text-based system.

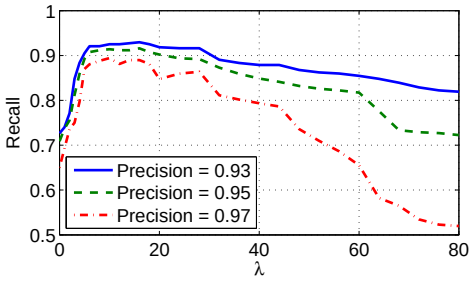

Figure 8: The recall at different precisions for the hybrid scheme using varying values of λ.

seconds which is required for all systems. The recognition latency for the text-based recognition pipeline is 0.22 seconds per spine image. The recognition latency for the image feature-based recognition system is 0.57 seconds per spine image. We evaluate the latency using our query images, which contains an average of ∼10 books per query. While processing the spines in parallel, the average recognition latency is 1.24 seconds.

4.2 Text-based Spine Recognition Evaluation

We further evaluate the performance of the text-based recognition pipeline which was not considered in previous works. We show in Fig. 9 the recall performance of the top scoring title for different resolutions using different scoring schemes. We observe for all different image sizes, using simple exact word matching scheme performs worst. Using the NN word matching to match the keyword to the dictionary helps improve the recognition performance by ∼8% for all different resolutions. With the additional *tf-idf* weighting, another ∼8% improvement can be observed. That is, a total of ∼16% improvement on recall can be gained by choosing a better scoring scheme.

The word recognition performance of the correctly identified titles is shown in Fig. 10 for exact word matching and NN word matching. We only consider words that have three or more characters. Using NN word matching, we in effect do spell checking and achieve a recognition performance improvement of ∼9% over exact word matching.

The word recognition performance drops when the resolution is lowered. However, we observe that the recall performance of the top scoring title improves no more than 5% when the spine height is above 768 pels. This suggests that SVGA sized images are sufficient when network bandwidth is scarce.

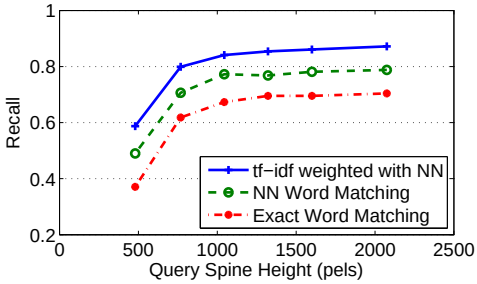

Figure 9: The recall performance of the top scoring book spine for three spine text database search schemes.

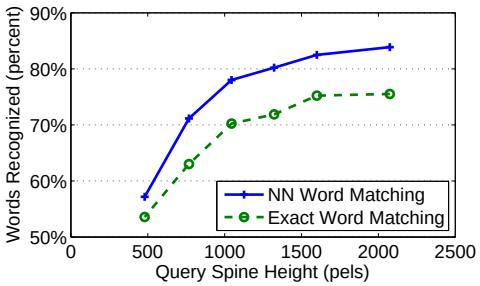

Figure 10: Word recognition performance of the correctly identified titles for different spine resolutions.

5. CONCLUSIONS

We have presented a new hybrid recognition system for identifying books on a bookshelf for use in book management systems. From a query image of a bookshelf, we extract the individual book spine images. Text is detected from the book spine images and used as keywords to search through a book spine text database. The text-based recognition system achieves high recall but low precision. We extract image features from the book spine images and use them to match the spines to a book spine image database. The image feature-based recognition system achieves a moderate recall at high precision. We combine the text-based recognition system with the image feature-based recognition system to form a hybrid recognition system. Through extensive experiments, we demonstrated that the hybrid scheme achieves a significantly higher recall of 91%, when the image feature-based and text-based systems have recall of 72% and 40% respectively.

6. REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3), 2008.

[2] D. Chen, S. Tsai, K.-H. Kim, C.-H. Hsu, J. P. Singh, and B. Girod. Low-cost asset tracking using location-aware camera phones. Number 1, San Diego, California, USA, 2010.

[3] D. M. Chen, S. S. Tsai, B. Girod, C.-H. Hsu, K.-H. Kim, and J. P. Singh. Building book inventories using smartphones. In *Proc. ACM Multimedia (MM'10')*, MM '10, Firenze, Italy, 2010. ACM.

[4] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen., V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *International Conference on Image Processing*, 2011.

[5] D. Crasto, A. Kale, and C. Jaynes. The smart bookshelf: A study of camera projector scene augmentation of an everyday environment. In *Proc. IEEE Workshop on Applications of Computer Vision (WACV'05)*, Breckenridge, CO, January 2005.

[6] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 1981.

[7] D. Lee, Y. Chang, J. Archibald, and C. Pitzak. Matching book-spine images for library shelf-reading process automation. In *Proc. IEEE International Conference on Automation Science and Engineering (CASE'08)*, Arlington, VA, September 2008.

[8] M. Loechtefeld, S. Gehring, J. Schoening, and A. Krueger. Shelftorchlight: Augmenting a shelf using a camera projector unit. *UBIProjection 2010 - Workshop on Personal Projection*, 2010.

[9] K. Matsushita, D. Iwai, and K. Sato. Interactive bookshelf surface for in situ book searching and storing support. In *Proceedings of the 2nd Augmented Human International Conference*, New York, NY, USA, 2011.

[10] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, June 2006.

[11] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, Anchorage, AL, June 2008.

[12] N. Quoc and W. Choi. A framework for recognition books on bookshelves. In *Proc. International Conference on Intelligent Computing (ICIC'09)*, Ulsan, Korea, September 2009.

[13] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5), 1988.

[14] I. H. Witten, A. Moffat, and T. C. Bell. Managing gigabytes: Compressing and indexing documents and images. 1999.

[15] T. Yeh and B. Katz. Searching documentation using text, ocr, and image. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 2009.