Mobile augmented reality for books on a shelf

Conference Paper · July 2011

DOI: 10.1109/ICME.2011.6012171 · Source: DBLP

MOBILE AUGMENTED REALITY FOR BOOKS ON A SHELF

David Chen[1], *Sam Tsai*[1], *Cheng-Hsin Hsu*[2], *Jatinder Pal Singh*[2], *Bernd Girod*[1]

[1] Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA
Email: {dmchen, sstsai, bgirod}@stanford.edu
[2] Deutsche Telekom R&D Laboratories USA, Los Altos, CA 94022, USA
Email: {cheng-hsin.hsu, j.singh}@telekom.com

ABSTRACT

Retrieving information about books on a bookshelf by snapping a photo of book spines with a mobile device is very useful for bookstores, libraries, offices, and homes. In this paper, we develop a new mobile augmented reality system for book spine recognition. Our system achieves very low recognition delays, around 1 second, to support real-time augmentation on a mobile device's viewfinder. We infer user interest by analyzing the motion of objects seen in the viewfinder. Our system initiates a query during each low-motion interval. This selection mechanism eliminates the need to press a button and avoids using degraded motion-blurred query frames during high-motion intervals. The viewfinder is augmented with a book's identity, prices from different vendors, average user rating, location within the enclosing bookshelf, and a digital compass marker. We present a new tiled search strategy for finding the location in the bookshelf with improved accuracy in half the time as in a previous state-of-the-art system. Our AR system has been implemented on an Android smartphone.

Index Terms— Mobile Visual Search, Mobile Augmented Reality, Book Spine Recognition

1. INTRODUCTION

Many visual search applications [1, 2, 3, 4] now enable mobile devices to retrieve information about products simply by snapping a photo. Typically, robust image-based features like SIFT [5], SURF [6], or CHoG [7] are extracted from the photo and matched against an online database, yielding accurate retrieval results even in the presence of photometric and geometric distortions.

There is also growing interest in applications that continuously augment the mobile device's video viewfinder with relevant information about the objects currently visible. Existing augmented reality (AR) applications use a smartphone's camera, digital compass, and Global Positioning System (GPS) sensors to create virtual layers on top of building facades and product packages seen in the phone's viewfinder [8, 2, 9, 10]. Low-latency, robust augmentation is often achieved using a combination of server-side visual search and client-side visual tracking.

In this paper, we develop a mobile AR system for a class of objects not considered in previous AR systems: book spines. Automatic book spine recognition is useful for generating an inventory of books and for retrieving information about a book without taking it off the bookshelf. However, identifying individual book spines in a bookshelf rack photo is challenging because each spine has a small area relative to the whole image and other spines act as clutter. Lee et al. [11] quantize each spine's colors and matches spines by color indices. Quoc and Choi [12] segment spine regions and extract titles by optical character recognition (OCR). Crasto et al. [13] deploy a calibrated projector-camera system to track the books that are removed from a shelf. In our previous work [14, 15], we used a combination of line-based spine segmentation and feature-based image retrieval to recognize book spines which are photographed in arbitrary orientations and under various lighting conditions. These prior systems all have recognition latencies of at least several seconds, making them less suitable for real-time mobile AR.

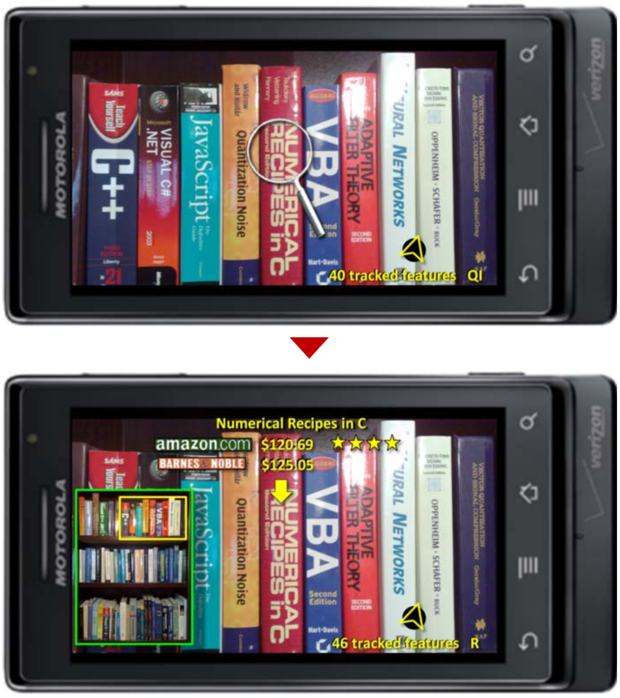


Fig. 1: Our new mobile augmented reality system for book spine recognition. (Top View) The user points the magnifying glass at a particular book spine. (Bottom View) About 1 second later, the viewfinder is augmented with the book's title, prices from competing vendors, an average user rating in stars, and a yellow box highlighting the location in the larger bookshelf. The digital compass arrow in the lower right-hand corner continuously shows the direction in which the phone is pointing. Demo video: `http://www.youtube.com/watch?v=fWOw2K1TzFk`

As depicted in Fig. 1, our new mobile AR system enables a user to point the camera at a book spine and see the book's title, prices from competing vendors, and an average user rating augmented in the video viewfinder after about 1 second. Optionally, images of the book's front and back covers can also be shown in the viewfinder to provide more information about the book. To show the location of the books currently visible in the viewfinder, we provide two visual aides: (1) a thumbnail of the surrounding bookshelf is displayed on the left side of the viewfinder and a yellow box highlights where the books are placed in the bookshelf, and (2) a digital compass arrow is drawn in the lower right-hand corner indicating the direction in which the phone is pointing. As another possible augmentation, our system plays an audio review of the book using the phone's text-to-speech function.

On the mobile device, the motion of objects seen in the viewfinder is analyzed to detect periods of low motion, when the user is likely interested in the contents of the viewfinder. At the start of each low-motion interval, a new query is triggered. Since user interest is automatically inferred, there is no need to press a button to initiate a query. This selection mechanism also has the positive effect of avoiding query frames severely degraded by motion blur, which occur when the user rapidly moves the phone. On the server, the spines are segmented from the query frame, and each spine is efficiently matched against a database of spine images by vocabulary tree scoring [16] and RANSAC-based geometric verification [17] on a shortlist of database candidates. To determine the precise location within the surrounding bookshelf, the query frame which shows spines on a single rack is matched against an image which shows the whole bookshelf.

The remainder of the paper is organized as follows. Sec. 2 gives background on line-based spine segmentation and feature-based spine recognition algorithms. Then, Sec. 3 presents our new mobile AR system, introducing our intuitive user interface, explaining how a user's intent to focus on a new book spine is inferred from the motion of objects seen in the viewfinder, and describing how fast rack-to-shelf matching is performed through a tile-based search scheme. Experimental results in Sec. 4 show the performance and advantages of our methods. Finally, Sec. 5 concludes the paper.

2. MOBILE BOOK SPINE RECOGNITION

In the book spine recognition system of [15], the user has to press a button on the mobile device to initiate a new query. A photo is taken by the onboard camera and transmitted over a wireless network (e.g., WLAN, 3G, 4G) to a server which contains a large database of labeled book spines. Matching the query photo directly against the database of spines yields poor retrieval results, because the spines in the query photo act as clutter toward one another. Thus, the spines in the query photo are first segmented by detecting edges and finding long, straight edges of similar orientation, corresponding to the boundaries between book spines. Then, robust image-based features are extracted from the individually segmented spines and matched against the database of spines using vocabulary tree scoring [16] and RANSAC-based geometric verification [17] on a shortlist of database candidates. The recognized spines' identities and boundaries are sent back to the mobile device and displayed on the viewfinder.

Compared to the system reported in [15], our new AR system has several new features and important advantages:

- There is no need to press a button, as user interest is automatically inferred by analyzing the motion of objects shown in the viewfinder.
- Recognition latency is reduced from about 3 seconds in the previous system to about 1 second in the new system, by quickly selecting a query frame from viewfinder frames at the start of a low-motion interval.
- The location of the current books is highlighted in a thumbnail of the bookshelf in the viewfinder, whereas the previous system just stored this location on the server. Finding the location is also made faster through a new tile-based search scheme.

With these improved features, the new AR system supports substantially greater interactivity and faster response.

3. MOBILE AUGMENTED REALITY SYSTEM FOR BOOK SPINE RECOGNITION

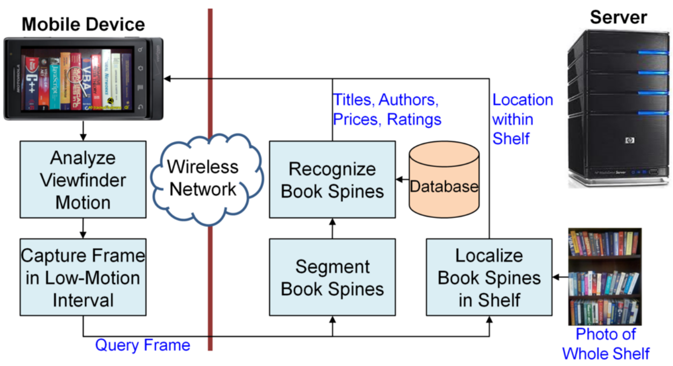


Fig. 2: Block diagram of our mobile augmented reality system.

A block diagram of our mobile AR system is drawn in Fig. 2. On the mobile device, motion analysis is performed on viewfinder frames, and a query frame is captured during each low-motion interval and transmitted to a server. On the server, to identify the book spines shown in the query frame, the spines are segmented and recognized using the methods of [15] as summarized in Sec. 2. The titles, authors, prices, and ratings of recognized spines are retrieved from a database and sent back to the mobile device. Meanwhile, feature-based image matching between the query frame and a photo of the whole shelf previously taken enables us to precisely determine the location of the book spines in the surrounding shelf. Coordinates representing the location of the books are also sent back to the mobile device.

3.1. Motion Analysis for Initiating Queries

As the user rapidly moves the smartphone, the user is most likely not interested in the viewfinder's contents during this *high-motion* period. Conversely, during a *low-motion* period, the user is likely interested in the viewfinder's contents. Our system initiates a new query at the beginning of each low-motion period by uploading a viewfinder frame of 640×480 pixels to the server. Among all the recognized book spines, the center-most spine has its information augmented in the viewfinder.

The speed at which the smartphone is moving can be reliably estimated by the motion of objects seen in the viewfinder. This motion is computed by extracting and tracking Rotation Invariant Fast Features (RIFF) [10] from viewfinder frames captured at 15 Hz. We demonstrate our motion analysis technique on two test viewfinder

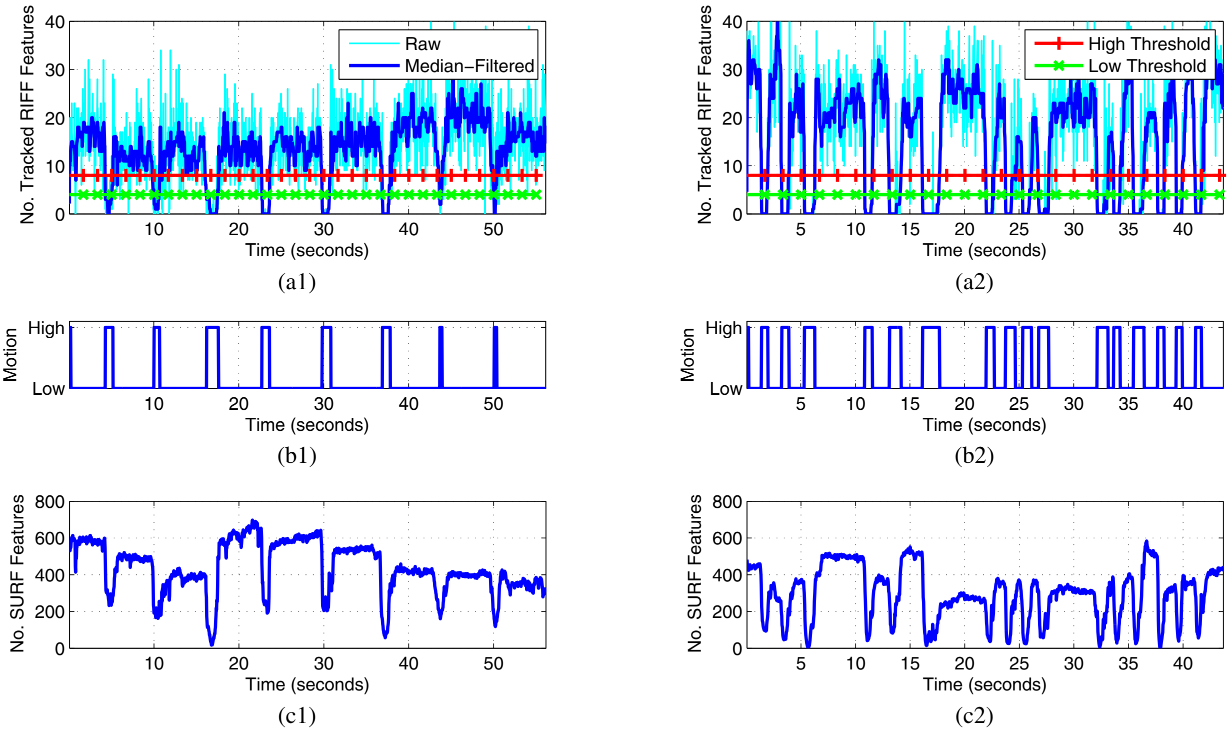


Fig. 3: Statistics for two different viewfinder sequences. (a1, a2) Number of tracked RIFF features between viewfinder frames. (b1, b2) Classification of motion into low and high states. (c1, c2) Number of SURF features for viewfinder frames.

sequences[1,2] captured with a Motorola Droid smartphone. The first sequence contains 9 different low-motion intervals, separated by 9 different high-motion intervals. Within each low-motion interval, there is a fair amount of hand jitter. The second sequence contains 17 different low-motion intervals, and most of them are shorter in duration than those in the first sequence.

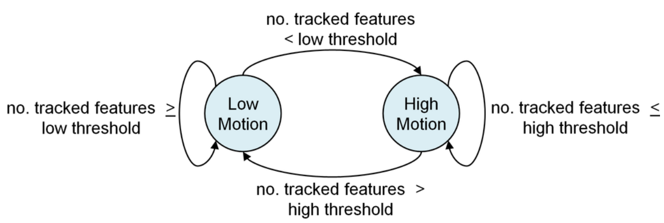
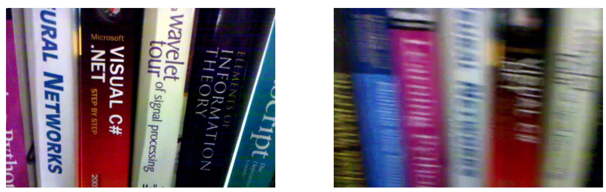


Fig. 4: Finite state machine for determining how to transition between low-motion and high-motion states on the mobile device.

Fig. 3(a1,a2) show traces of the number of tracked RIFF features for both sequences. Since the raw trace is very noisy, a median filter with a window of 7 samples is applied for more stable motion estimation. If $R[k]$ denotes samples in the raw trace, samples in the median-filtered trace are given by $M[k]$ = median $\left(\{R[k+\delta]\}_{\delta=-3}^{3}\right)$. Since $M[k]$ depends on future samples $\{R[k+\delta]\}_{\delta=1}^{3}$ and the samples are collected at 15 Hz, a small delay of 200 milliseconds is incurred compared to directly using $R[k]$. RIFF uses FAST corner keypoints [18] whose repeatability



(a) (b)

Fig. 5: Viewfinder frames selected from (a) low-motion and (b) high-motion intervals.

decreases sharply when there is motion blur, so a low (high) number of tracked features indicates a period of high (low) motion. A low (high) threshold is determined so that the number of tracked features during high-motion (low-motion) intervals lie below (above) the low (high) threshold. Subsequently, we use the finite state machine (FSM) in Fig. 4 to switch between low-motion and high-motion states. Having two thresholds instead of one is important to prevent rapid switching between states in a short duration due to noise, and the distance between the low and high thresholds is scaled in relation to the standard deviation of the noise in the median-filtered trace. The motion classifications given by the FSM are plotted in Fig. 3(b1,b2).

Fig. 5 shows two frames, one seleced from a low-motion interval and the other from a high-motion interval. As can be observed, the low-motion frame has more clearly defined details, while the high-motion frame suffers from motion blur which can severely degrade the line-based spine segmentation and feature-based spine recognition methods. Fig. 3(c1,c2) show traces of the number of SURF [6]

[1]`http://www.youtube.com/watch?v=9Py1Q0jz6DQ`
[2]`http://www.youtube.com/watch?v=RpGtpLOikdk`

features in both test sequences. During each high-motion period, there is a significant drop in the number of SURF features due to motion blur. A frame with few SURF features is likely to yield an inaccurate image retrieval result. Thus, our choice to initiate a query during a low-motion interval not only corresponds to a period of very probable user interest, but also avoids selecting useless blurry frames.

3.2. Fast Tiled Search for Rack-to-Shelf Matching

As a user identifies books with our AR application, an inventory program on the server records all the books queried by the user. The inventory information currently includes (1) location-agnostic details such as the book titles, authors, prices, user ratings, and reviews, and (2) location-aware details such as the direction that a person should be facing in a room to see the books and the specific position of a set of books within the surrounding bookshelf. When a query is initiated, we compute the phone's direction from the onboard magnetic field sensors. The estimated direction is shown as a digital compass arrow in the lower right-hand corner of the viewfinder (see Fig. 1). In this section, we focus on the more challenging problem of precisely locating books within the surrounding bookshelf. Note that the methods discussed in this section are not used to recognize the individual book spines in a query viewfinder frame; spine recognition is performed using the vocabulary tree scoring and RANSAC-based geometric verification methods described in Sec. 2.

To localize the books currently visible in the viewfinder within the surrounding bookshelf, two types of approaches are possible: (1) location estimation based on a recent trace of the accelerometer readings and knowledge of an anchor point, and (2) location estimation based on matching the viewfinder frame against an image of the whole bookshelf. Both approaches have been previously evaluated [15], and the image-based approach has been found to give noticeably higher localization accuracy. In this section, we describe a new image-based localization strategy that is faster and more accurate than the method in [15].

Before querying individual books, the user takes a 960×1280 photo that shows the entire bookshelf (e.g., Fig. 6(a)). This bookshelf photo I_{shelf} can be repeatedly reused for localization purposes, even if a small number of books are subsequently removed from or misplaced in the bookshelf. I_{shelf} is only retaken when we focus on a new shelf or when the contents of the current shelf change significantly. Each 640×480 query frame I_{query} (e.g., Fig. 5(a)) shows a particular rack in the shelf. Feature-based image matching between I_{query} and I_{shelf} allows us to precisely localize where the spines in I_{query} reside within the whole bookshelf shown in I_{shelf}. Note that I_{shelf} is used for localization only and is not used to recognize individual spines in I_{query}.

The system in [15] used all the local feature descriptors in I_{shelf} to build a k-d tree. For each descriptor in I_{query}, the first and second nearest descriptors in I_{shelf} are found by searching the k-d tree, and a tentative match is formed with the first nearest descriptor in I_{shelf} if a distance ratio test is passed [5]. Tentative matches are then verified using RANSAC with an affine model. We refer to this scheme as *Full Search*.

Although I_{query} covers only a portion of I_{shelf}, Full Search potentially compares every descriptor in I_{query} to every descriptor in I_{shelf}. Thus, many descriptors in I_{shelf} act as outliers, making the matching process less accurate and slower. We address this problem with a new *Tiled Search* strategy, depicted in Fig. 6(b). First, long nearly horizontal edges are detected in I_{shelf} to find the boundaries between racks in the bookshelf. Second, each rack is split into C_{rack} nonoverlapping tiles of equal width, where C_{rack} is an adjustable system parameter. Fig. 6(b) illustrates a sample 3-rack bookshelf with $C_{\text{rack}} = 2$ tiles per rack. For each tile, all the descriptors falling within that tile are used to build a k-d tree specific to that tile. Next, we exploit the fact that consecutive query frames tend to cover different portions of the same rack. If the previous query frame was matched to a tile in the i^{th} rack, for the current query frame, we first search the tiles in the i^{th} rack and terminate the search if the number of post-RANSAC inliers exceeds a threshold T_{RANSAC}; no false positive image matches are ever observed for a sufficiently high value of T_{RANSAC}. Only if fewer than T_{RANSAC} inliers are found does the search continue into tiles in the other racks. As we will show in Sec. 4.3, Tiled Search significantly reduces the rack-to-shelf matching latency while actually giving a slight boost in matching accuracy compared to Full Search. We will also show empirically the tradeoff between search latency and number of feature matches as the parameter C_{rack} is varied.

(a) (b)

Fig. 6: (a) Image of the whole bookshelf with feature keypoints overlaid. (b) Same image split into $C_{\text{rack}} = 2$ tiles per rack.

4. EXPERIMENTAL RESULTS

4.1. Recognition Latency

In this section, we report the performance of our new AR system and show it has much lower recognition latency than the system reported in [15]. Both systems use a Motorola Droid smartphone running Android 2.1 on a 550 MHz processor. The recognition server has a 3.2 GHz processor. This server performs line-based spine segmentation, extraction of upright SURF features [6], vocabulary tree scoring [16] with a set of 1 million visual words, soft binned quantization [19], and RANSAC-based geometric verification [17] on a shortlist of the top 50 candidates out of a database of 2148 labeled book spines. Query viewfinder frames are uploaded over a WiFi network with 1 Mbps transfer rate; our system would very likely be deployed in a library, bookstore, office, or home with a WiFi network.

Fig. 7 compares the latencies for different operations in the previous system [15] and our new AR system. Both systems are tested on a set of 40 rack images (all 640×480 resolution) which are available online[3]. This collection also includes a 960×1280 image showing the entire surrounding bookshelf, where the shelf contains all the books shown in the 40 rack images. Book spines are photographed in different orientations and under different lighting conditions.

[3] http://tinyurl.com/3k9skw2

First, for image capture, the previous system initiates a photo capture operation after the user presses a button, a process that takes 2 seconds on average. When the camera shutter closes during photo capture, the viewfinder screen also turns black momentarily, which is an undesirable effect for continuous AR. In contrast, our AR system captures a viewfinder frame at the beginning of a low-motion period, taking 200 milliseconds to collect enough samples for the median-filtered trace and 100 milliseconds to copy a query frame into an upload buffer, with no interruption of the viewfinder stream. Second, the latencies for image upload, line-based spine segmentation, and feature-based spine recognition are similar in the two systems. Then, the rack-to-shelf matching method is faster in our new system because we use a more efficient Tiled Search compared to the Full Search used in the prior system. In total, recognition latency is reduced from about 3 seconds in the previous system to about 1 second in the new system. The low latency of the new system is very important for supporting real-time AR. Interestingly, since queries are triggered automatically, rather than by a conscious user input, the remaining 1 second latency is hardly noticeable. Recognition results "magically" appear, as soon as the user hovers over the book spine of interest.

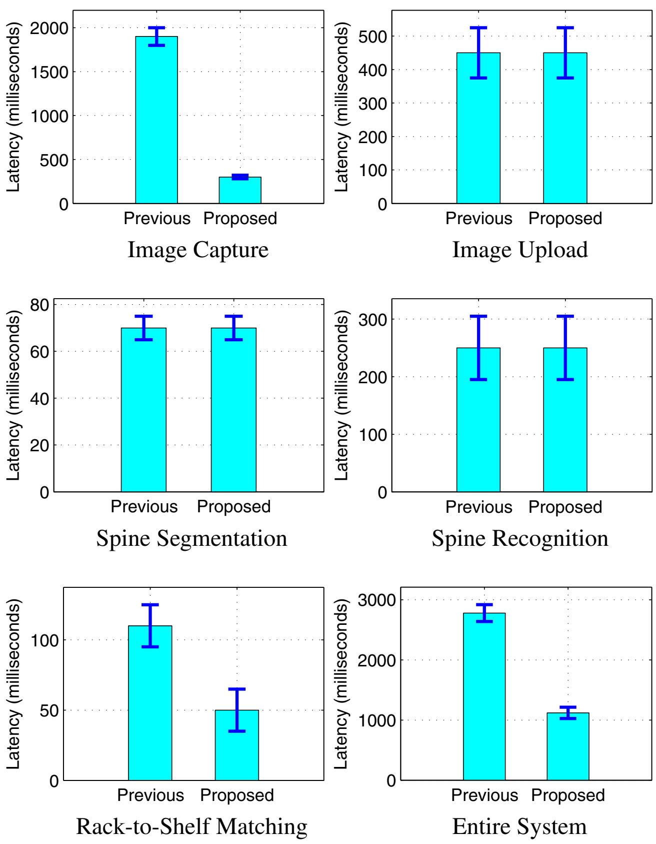


Fig. 7: Comparison of latencies for different operations between the previous system [15] and our newly proposed system. The error bars indicate standard deviations.

4.2. Recognition Accuracy

For book spine recognition, we use the retrieval system described in [15]. Each query spine in the aforementioned 40 test rack images is matched against the database of 2148 labeled spines, by vocabulary tree scoring and RANSAC-based geometric verification on a short-list of 50 database spine candidates. If at least $T_{\text{RANSAC}} = 25$ feature matches are found between the query spine and the best database candidate, a good match is deemed to be found and the information for that matching database spine is retrieved to be displayed on the phone's screen. With these settings, we achieve 80 percent recall and 95 percent precision in identifying all the spines shown in the 40 test rack images. To avoid returning false positives to users, it is important to attain high precision at the expense of slightly lower recall.

4.3. Rack-to-Shelf Matching

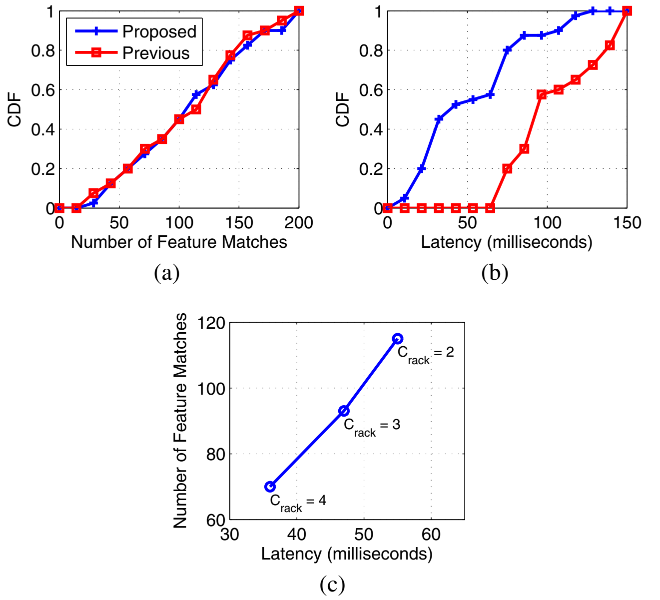


Fig. 8: Statistics for localizing query book spines within an image of the whole shelf. (a) Cumulative distribution function (CDF) for number of feature matches. (b) CDF for search latency. (c) Number of feature matches versus search latency, as the number of tiles C_{rack} per rack is varied.

In Sec. 3.2, we described the Full Search and Tiled Search methods for matching a query image to an image of the entire bookshelf. For the same 40 test images, the distribution of the number of feature matches between a rack image and the larger bookshelf images with $C_{\text{rack}} = 2$ is plotted in Fig. 8(a), where it can be seen that Tiled Search and Full Search perform comparably. Tiled Search obtains 115 features matches on average, slightly higher than the 111 matches obtained on average by Full Search, due to the avoidance of outliers in bookshelf regions distant from the current rack. In our design, Tiled Search will terminate whenever any particular tile in the bookshelf image matches the rack image with more than $T_{\text{RANSAC}} = 50$ post-RANSAC inliers. Due to this early termination option, Tiled Search significantly reduces the latency compared to Full Search, as shown in Fig. 8(b). On average, Tiled Search takes

54 milliseconds per query image compared to 107 milliseconds for Full Search.

The parameter C_{rack} can be adjusted to reduce search latency or increase the number of feature matches. Fig. 8(c) shows this tradeoff for $C_{\text{rack}} = 2, 3, 4$. Having fewer tiles per rack causes each tile to become wider, which increases the number of feature matches between a whole bookshelf image and a query frame, but also increases the image matching latency. We observe that using $C_{\text{rack}} = 4$ tiles still yields a decent number of feature matches while cutting the latency by 35 percent compared to $C_{\text{rack}} = 2$ tiles.

5. CONCLUSIONS

We have developed a new mobile augmented reality system for recognizing book spines. Our system achieves a very low recognition latency around 1 second, which is crucial for near instantaneous augmentation on the mobile device's viewfinder. There is no need to press a button to initiate a query, because user interest is automatically inferred from the motion of objects seen in the phone's viewfinder. In addition to augmenting the viewfinder with a recognized book spine's identity, we also highlight the location of the books in the surrounding bookshelf. Our book spine recognition system provides a fast way of retrieving information with a mobile device about books in a library, bookstore, office, or home, without ever taking a book off the bookshelf. Book spine recognition can be easily combined with book cover recognition to create a joint system that can recognize any facade of a book. Other potential applications of our mobile AR system include helping librarians reshelve misplaced books; aiding bookstore clerks in organizing books on shelves according to the books' subjects; and guiding an individual toward a particular book of interest in a library or bookstore.

6. ACKNOWLEDGMENTS

We thank Gabriel Takacs for sharing his RIFF code and helping us port it to the Android platform. We also thank the reviewers for their insightful comments, which were very helpful in improving this paper.

7. REFERENCES

[1] Google, "Google Goggles: use pictures to search the web," http://www.google.com/mobile/goggles.

[2] Kooaba, "Kooaba Visual Search: get instant product information," http://www.kooaba.com.

[3] Nokia, "Nokia Point and Find: tag places and objects," http://europe.nokia.com/services-and-apps/nokia-point-and-find.

[4] Amazon, "Amazon Remembers: create a visual list of products," http://www.amazon.com/gp/remembers.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, June 2008.

[7] V. Chandrasekhar, Y. Reznik, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "Quantization schemes for low bitrate compressed histogram of gradients descriptors," in *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, San Francisco, CA, USA, June 2010, pp. 33–40.

[8] Layar, "Layar Reality Browser: digital information on top of the real world," http://site.layar.com/download/layar.

[9] D. Chen, S. Tsai, R. Vedantham, R. Grzeszczuk, and B. Girod, "Streaming mobile augmented reality on mobile phones," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, Orlando, FL, USA, October 2009, pp. 181–182.

[10] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010, pp. 934 –941.

[11] D. Lee, Y. Chang, J. Archibald, and C. Pitzak, "Matching book-spine images for library shelf-reading process automation," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Arlington, VA, USA, September 2008, pp. 738–743.

[12] N. Quoc and W. Choi, "A framework for recognition books on bookshelves," in *Proc. International Conference on Intelligent Computing (ICIC)*, Ulsan, Korea, September 2009, pp. 386–395.

[13] 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)*, Breckenridge, CO, USA, January 2005, pp. 218–225.

[14] D. Chen, S. Tsai, C.-H. Hsu, K.-H. Kim, J. P. Singh, and B. Girod, "Building book inventories using smartphones," in *ACM International Conference on Multimedia (MM)*, Firenze, Italy, October 2010, pp. 651–654, ACM.

[15] 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," in *Applications of Digital Image Processing (ADIP) XXXIII*, San Diego, CA, USA, August 2010, p. 77980R.

[16] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, NY, USA, June 2006, pp. 2161–2168.

[17] 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*, vol. 24, no. 6, pp. 381–395, 1981.

[18] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision (ECCV)*, Graz, Austria, May 2006, vol. 1, pp. 430–443.

[19] 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)*, Anchorage, AL, USA, June 2008, pp. 1–8.