St. Joseph's Institute of Technology
3rd International Conference on Recent Trends in Engineering and Technology
4th and 5th May 2018

FEATURE MATCHING BASED BOOK DETECTION SYSTEM

[1]Sai Mukkundan R, [2]Melvin A, [3]Shoban Chander E and [4]Dr.V.Sathiesh Kumar
[1,2,3]Final year, Electronics and communication Engineering, Madras Institute of Technology, Chennai 600044
[4]Asst Professor, Department of Electronics and Communication, Madras Institute of Technology, Chennai 600044

***Abstract*--Finding a particular book from a large stack in a library is a difficult and time consuming task. Library users need to search the entire stack all by themselves and this is not a simple job. To reduce their strain and guide them in accessing the book they want, an autonomous system for book detection is proposed. This paper is all about the identification of user queried book from a huge collection of books in the library stack in an efficient way. The book to be detected is found by using image processing techniques like feature extraction and feature matching methods. It involves maintaining a separate database of book spine images locally and comparing the existing features of the queried book with the features present in the test image. DoG key point detector and RootSIFT feature descriptor are used to find the maximum likeliness between the queried and test image. The algorithm applied is an accurate method as it proves to be scale, affine and illumination invariant and can also be deployed easily in real-time at minimal cost**.

***Keywords*--Book Detection, Book Spine, DoG, RootSIFT, feature matching, RANSAC, Linear Slide.**

I. INTRODUCTION

THIS system aims to reduce the time taken in identifying a particular book from the library stack by using image processing techniques rather than using other conventional methods[1][2]. It can also be done using RFID tags and reader[1] or barcodes and barcode scanner[2], but deployment of such system is time consuming. Xunrui Duan et al.[3] proposed a method based on identification and extraction of call numbers from the libraries. But it failed for those books which had same background as that of the call number ,for books in which their call numbers are extended beyond their spine and they are prone to multi-scaled image capturing method. Kenji Iwata et al. [4] proposed a method for book cover identification using four directional feature extraction from a low resolution images. The only area this algorithm fails is when the book cover is damaged. Spencer G Fowers et al. [5] made a slight modification in the original algorithm by using colour DOG keypoint detection instead of normal one ,this increased computational complexity since the keypoint detection is performed in YCbCr colour space. But it cannot be employed in low cost microprocessors due to its computation load and it requires GPU for better functionality.

This method requires only a camera and a local database to identify the book. The idea of detecting the book is done by detecting the interest points and comparing the features around these interest points. Character segmentation and recognition involves a lot of training datasets, due to a wide range of fonts in books cover is not needed. Keypoint detector and local feature descriptors are used to identify unique, repeatable and precise keypoints. Images are matched using Feature Matching[6]. There is no need for character recognition or any hefty machine learning techniques for book detection.

II. METHODOLOGY

The prerequisites to be carried out,

1. Get the required book details from the user.
2. Locate the stack and start searching for the book.

Based on the user query, the image is picked up from the database and DoG[7] is applied to get scale invariant key points and RootSIFT[8] feature descriptor is applied to get scale and rotation invariant features. Similar operation is performed on test image to obtain another set of features. Features of both query and test image are compared, and a test frame from video of the stack with a maximum number of matched key points is taken and minimum number of matched key points is set as the threshold for rough classification of query book. Homography (transfer matrix) is formed from matched key points to which original points in query image are fed to obtain the book location in the test frame. The overall workflow is given in Fig.1.

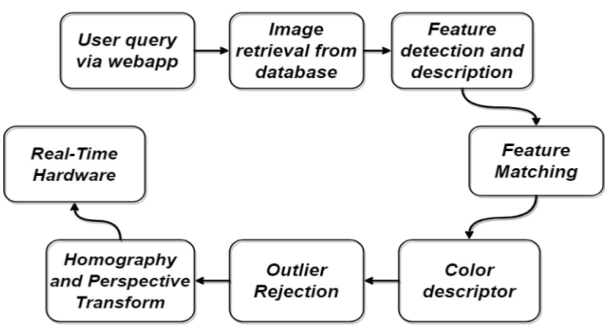

Fig 1. Overall Workflow

*This work was not supported by any organisation.
This work was done under the guidance of [4]Dr.V.Sathiesh Kumar, Department of Electronics and Communication, Madras Institute of Technology, Chennai 600044
[1]Sai Mukkundan R (mail: sai.mukkundan@gmail.com), [2]Melvin A (mail: melvindoss@gmail.com) and [3]Shoban Chander E (mail: shobanchander@gmail.com) are in Final year, Electronics and communication Engineering with Madras Institute of Technology, Chennai 600044

A. *User Query to Database*

The user query is given via Web-App as shown in fig 2, which is locally hosted on the library network using Apache Tomcat web container. Anyone connected to the library's network can send the book details to the moving Kiosk located inside the library. The book name, author name and the edition are sent via socket connection from the user side (client). The SBC (single board computer) running server code receives this info and picks the query image path from the database.

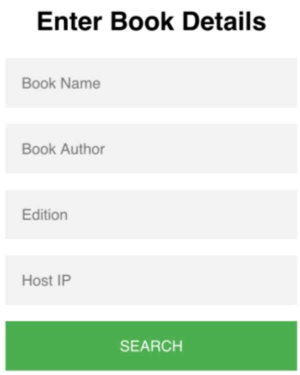

Fig 2. Web-Application (HTML form)

B. *Query Image Retrieval:*

The data from the socket connection is received and the appropriate book spine image for the user query is loaded from the database. The database consists of book name, author name, book edition and the local path of the image as its fields as shown in fig 3. The local image path is retrieved from the database. This image is now the query image upon which feature detection and description are performed.

S no	Book Name	Author	Edition	Book Path

Fig 3. Database fields

C. *USING FEATURE EXTRACTION AND MAPPING*

The steps followed in this technique are:

i. Identification of key-points and their descriptors and extracting the feature vectors out of the test frame and the image of the book in the database.
ii. Map the features using K nearest neighbours algorithm and Lowe's ratio test.
iii. Use RANSAC[9] (Random sample consensus) to remove the unwanted mappings.
iv. Colour feature extraction to distinguish between similar books.

1) *KEY-POINT IDENTIFICATION AND FEATURE EXTRACTION*

Key-points are the points of interest on an image viz, corners and edges. Each key-point that you detect has an associated descriptor useful when matching is done. The key-points are found out using DoG (Difference of Gaussian) and describe them using RootSIFT feature descriptor.

Difference of Gaussian (DoG) function is implemented by applying Gaussian blur on the original source image and subtracting it with the source image. DoG is same as LoG (Laplacian of Gaussian) in function but computationally cheap compared to it. The difference of Gaussian is computed at two nearby scales separated by a constant multiplicative factor k.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (1)$$

. The whole process is depicted in Fig 4 and Fig 5.1

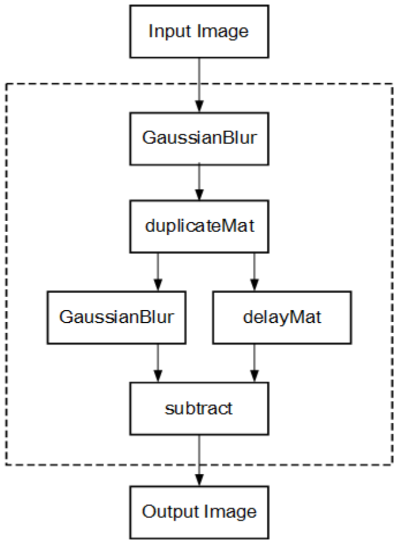

Fig 4. DoG (difference of Gaussian flow)

2) *LOCATE MAXIMA-MINIMA IN DoG IMAGES*

Maxima and minima of the DoG images are detected by comparing a pixel (marked with X) to its 26 neighbours as shown in Fig 5.2. If the intensity value (i.e., without sign) of all pixels in the DoG image are less than the centre most pixel, then the centre most point is detected as a key point which is scale invariant as DoG is performed on all scales. DoG is an oversimplification of LoG (2nd derivative edge detection filter) hence it detects corners and edges.

Setting an intensity threshold among the corners and edges, eliminates the low contrast edges and corners.

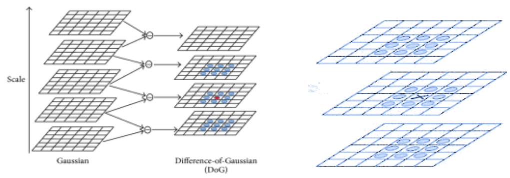

Fig 5.1, Fig 5.2 Scale space extrema points detection, Local maxima or minima extraction

D. *SIFT MAPPING*

SIFT aims to find highly distinctive locations in an image. The locations are not merely 2D locations but they are in the image's scale space and have three coordinates (i.e.) x, y, and scale. SIFT describes the local pixel intensity distribution around these key points which provides the distinctiveness between similar key points.

1) *LOWE's KEY-POINT DESCRIPTOR:*

i. Assign every key-point an orientation by calculating histogram of gradient orientations for every pixel in its neighbourhood and picking the orientation bin with the highest number of counts.
ii. Assign every key-point a 128-dimensional feature vector based on the gradient orientations of pixels in 16 local neighbourhoods as shown in fig 6.

DoG gives us scale invariance, step 1 gives us rotation invariance, and step 2 gives us a "fingerprint" of sorts that can be used to identify the key-point. Together they can be used to match occurrences of the same feature at any orientation and scale in multiple images.

- Compute a descriptor for the local image region about each key-point that is highly resistant to variations in viewpoint and illumination.
- Rotate the window to the dominant orientation and scale the window size based on the scale at which the point was found.

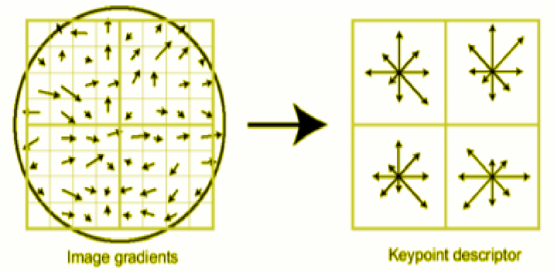

Fig 6. SIFT vector formation

E. ROOT-SIFT

When comparing histograms the Euclidean distance often yields inferior performance than when using the chi-squared distance or the Hellinger kernel[3]. The original 128-dim descriptor returned from SIFT is modified in such a way that while performing feature matching with Euclidean distance, it is actually measured with chi-squared distance.

Steps to modify SIFT to ROOTSIFT are:

i. Compute SIFT descriptors and L1-normalize each SIFT vector.

ii. Take the square root of each element in the SIFT vector. L2-normalize the resulting vector.

L2-normalization is better than L1-normalization as the error between the query and test descriptor are squared, such that only accurate matches are preserved by it after applying Lowe's Ratio Test. It's a simple extension, but this little modification increased the accuracy in feature matching between the test image and the query image.

$$S = \sum_{i=1}^{n} |y_i - f(x_i)|. \quad \text{L1 norm} \quad (2)$$

$$S = \sum_{i=1}^{n} (y_i - f(x_i))^2 \quad \text{L2 norm} \quad (3)$$

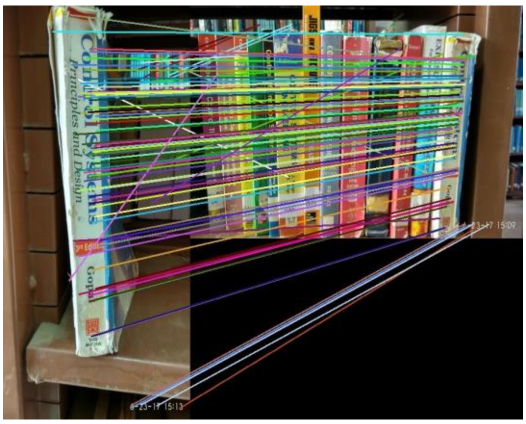

Fig 7. Feature Matching

F. FEATURE MATCHING

After obtaining the features from both query and test frame, they are compared for maximum matches by computing Euclidean distance between two 128-dim vectors and visualising it as shown in Fig 7. Corresponding points between two images is found by Brute-Force approach where a single descriptor in query image is compared with all the descriptors in the test image. K nearest neighbour matches two nearest neighbour in the test image for a single descriptor in the query image. Threshold for minimum number of matches is set as 18 and images with less matches shows that the book is not in the frame.

G. COLOUR DESCRIPTOR:

To distinguish books with similar text and logo, colour feature vector is added next in the pipeline. Construct a 9X9 rectangular window around every matched keypoint. Both images are converted into HSV plane and a mask is applied to these rectangular regions alone as shown in Fig 8 & Fig 9.Hue histogram is constructed only for those masked regions with 18 bins. This forms a feature vector, extraction of feature vectors from both test and query image is done followed by comparison using Euclidean distance. Finally a threshold is set before sending the image down the processing pipeline.

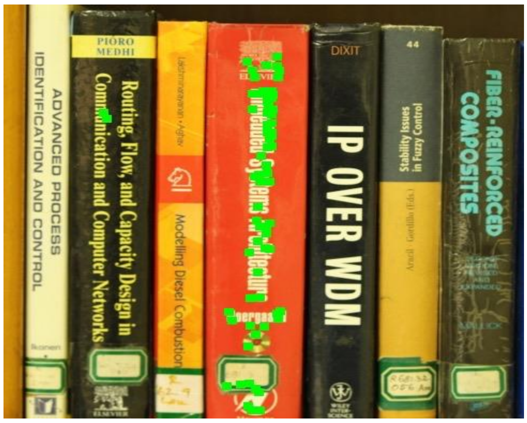

Fig 8. Window around matched keypoints

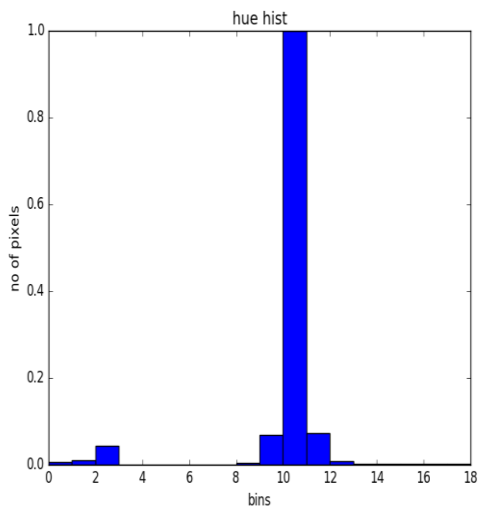

Fig 9. HUE histogram

The possible wrong matches are eliminated by adding the colour component alone to feature matching process. Colour descriptor is done to distinguish between different colour books from same publisher. It improved the overall accuracy of the system with reduced false positive count.

H. OUTLIER REJECTION

Lowe's Ratio is applied to further remove outliers. In case of an test image with no query book , there are completely random key points, and the nearest neighbour matching will not work since all keypoints are random .If the book to be detected is in the test frame, then the two nearest neighbours would be placed far apart in distance since the keypoints are distinct. Lowe's ratio of 0.7 is applied to eliminate false matches as shown in Fig 10.

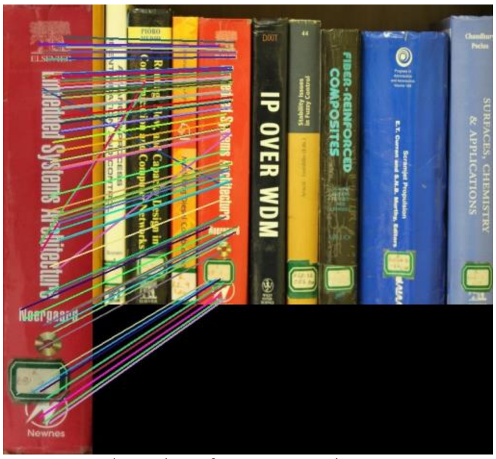

Fig 10. Good matches after Lowe's ratio test

I. HOMOGRAPHY AND PERSPECTIVE TRANSFORM

The matches obtained from Lowe's ratio test are passed through a line fitting algorithm called as RANSAC (Random Sample Consensus) which is mainly used to remove the outliers present in the matched features list.

1) HOMOGRAPHY

Homography matrix is just the transfer function between two different image planes. This can be computed provided correct corresponding points in both test and query image. Homography fails if the matches are all erratic and mostly wrong. Outliers are removed first, before homography is obtained.

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{4}$$

H is the homography matrix, (x1, y1) and (x2, y2) are the coordinates of matched keypoints in test image and query image.

2) RANSAC

RANSAC randomly selects a subset data points and tries to fit it in a desired model. RANSAC requires a data model, cost function and optimisation technique to perform effectively. The data model of interest is the transformation matrix between different image planes. Transformation matrix is obtained from the matched key points of query and test image. RANSAC selects randomly four matched key points, computes homography and categorises other points into inliers (sample consensus set) and outliers based on the reprojection threshold set by the user which determines that whether a data point can fit into an existing model or another four set of samples are taken to construct a new model, it is repeated for minimum number of iterations set by the user till a best fit data model with least average error occurs.

The cost function for above model is

$$s_i \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{5}$$

Destination points ***Si****(points in test frame) - (Homography matrix X Source points(****Xi, Yi****)) << Reprojection Threshold*

to qualify a key point as an inlier. Correct and accurate matches after applying RANSAC as shown in Fig 11.

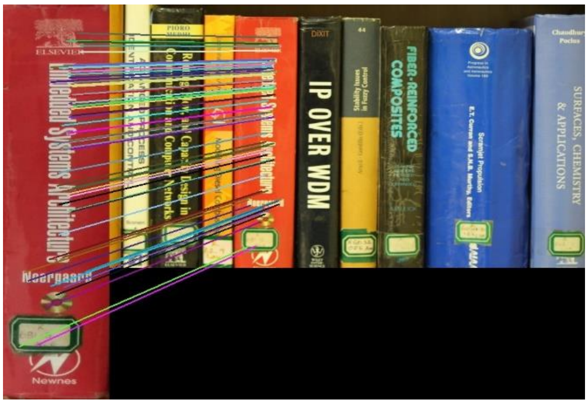

Fig 11. Correct matches obtained after RANSAC

3) PERSPECTIVE TRANSFORM

Provided a transformation matrix along with source points of one image plane, the destination points in another image plane can be obtained. Homography matrix of previous stage along with the corner points of book spine image from the database is multiplied to get the corresponding four points of book corners in the test frame. A bounding box is drawn to indicate the location of the book in the test frame as shown in Fig 12.

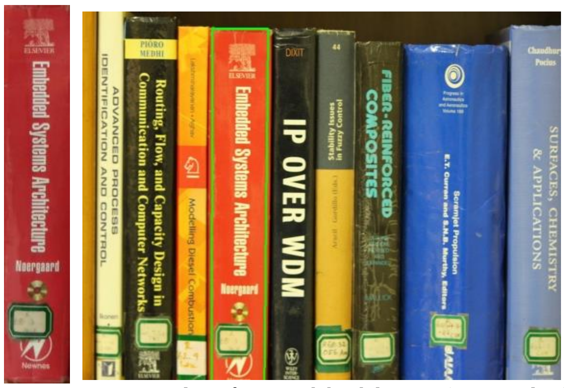

Fig 12.Query image and test frame with book location(green outline).

J. REAL TIME HARDWARE

The idea of locating desired book from the stack is modelled as a moving kiosk[10] hardware built from metallic frames, gears, shafts and wheels. The moving kiosk can lift the camera upto 900 mm covering three rows of stack easily. Linear slide is the perfect mechanism for lifting the camera to three different heights in a more precise and stable way as shown in Fig 13. The kiosk is built from five basic metallic frames of 300 mm in length with a footprint of 434 X 434 mm as shown in Fig 14. Metallic sliders are fitted on the sides of three metallic frames, in which individual frames can be moved by utilising pulley and thread mechanism.

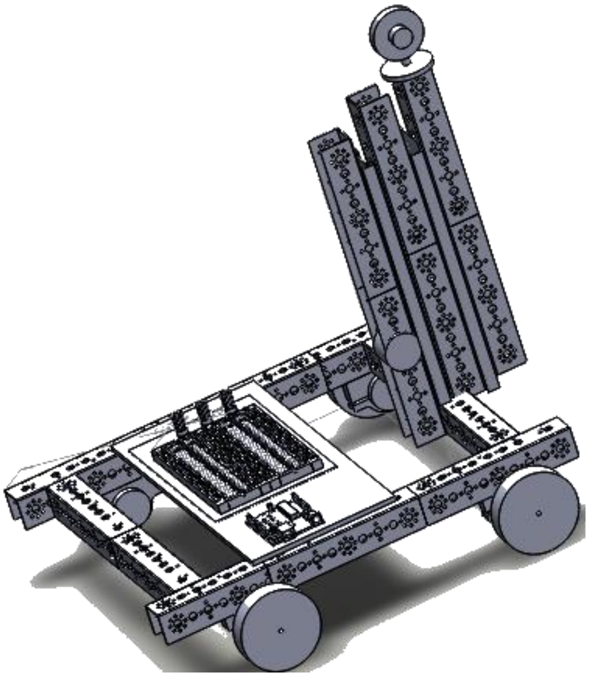

Fig13. Linear Slide Mechanism implemented in Kiosk (Isometric View)

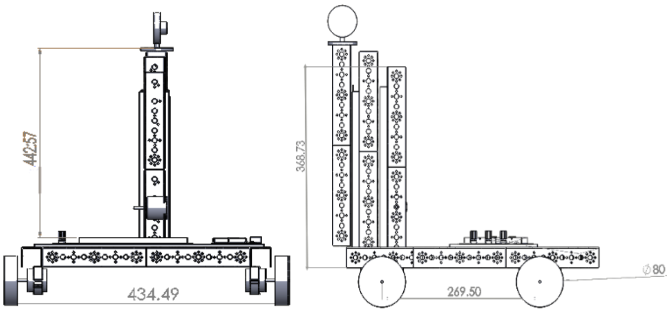

Fig 14. Kiosk Dimensions (Front View and Side View)

After mechanical aspect, movement of linear slide is maintained by ultrasonic sensors for row height (mounted on the metallic frame) determination and end of stack detection (mounted on the side of kiosk). The algorithm is made to run on a low power microcontroller like Raspberry pi 3 which serves as the master controller which communicates with the web-application locally and processes the images and instructs the secondary controller preferably Arduino for kiosk movement. The kiosk on receiving the query from the web app remotely starts to follow a black line with the help of IR sensors and DC motors while simultaneously processing the image obtained from the webcam. Raspberry pi 3 processes the images from webcam while the kiosk in motion. On successful book detection, Raspberry pi 3 instructs Arduino serially to stop the kiosk, shows the user the location of book and moves the kiosk back to the initial position along the same line. Real time performance in processing the image is achieved by using multithreaded programs in pi 3.Thus real time hardware can be easily deployed and book of any kind can be detected if the local database is updated with latest book information available in the library.

III. Experiments And Results

A. EXPERIMENTAL DATA

To test the accuracy of the algorithm, around 150 images of different book spine were gathered and feed into the locally hosted database. These images are captured in a high-resolution camera and are resized to a fixed height of 720 pixels. Before resizing the images are cropped to match the book boundaries. A separate database for this 150 images is created consisting of book name, author name, and edition along with image path as fields. Then a test dataset of 50 images of different stacks filled with books were taken for comparison as shown in Fig 15.The queried book image is compared to 50 test images for possible matches. Various feature detection and descriptor algorithms were applied and their accuracy is tabulated.

Fig 15. Dataset Preparation

B. EXPERIMENT RESULTS:

(*With Lowe's Ratio as 0.7 and minimum matches threshold at 3 and RANSAC Reprojection threshold as 4.0*)

Key-point Detection Algorithm	Matched key points	False Positives	Accurate Matches	Score	True Negatives
DoG	82	13	69	84	No
	22	10	12	54	Yes
SURF	8	8	0	0	No
MSER	4	0	4	100	No
STAR	4	0	4	100	No
HARRIS	4	0	4	100	Yes
	4	4	0	0	No
DOG with Colour descriptor	82	9	32	80	No

Table 1. Keypoint algorithm accuracy

From Table 1, it is clear that DoG with RootSIFT descriptor produces more matches than when other keypoint detectors are used. Even though STAR and MSER have the highest accuracy, they produce very few matches and with those few points computation of homography matrix is difficult. There are instances where DoG gives false readings for books of the identical publisher, the same logo in the book spine causes this error and it can be eliminated by combining colour descriptor before computing the transfer matrix. As seen from the table, the number of accurate matches from matched key points for a colour descriptor is reduced after applying RANSAC, but these matches are very sharp matches when compared to DoG applied alone.

Fig 16. Real time implementation.

IV. CONCLUSION

In this system, feature detection and matching were performed to identify the location of the book in the stack. Usage of colour descriptor for differentiating identical books differing in only by some words. Furthermore, it can be deployed it in real time using low power SBC like raspberry pi 3 encapsulated in a moving kiosk around the library. A user interface was created, a web app to communicate with the kiosk remotely. By using this system, time wasted in searching the book from a large library stack is saved drastically. The image processing algorithm was able to detect the book even if the book was occluded a bit, tilted, rotated and illuminated not uniformly as shown in Fig 16.

V. FUTURE WORK

The only side this system has to be improved is the procedure involved in updating the database when new books arrive at the library and moreover frequent updating and maintaining of large database of books manually is difficult and it can be improved by using some machine learning techniques like neural network to segment only the text region from the book spine image and to update the database fields automatically based on the classification result.

REFERENCES

[1]. Mohd.Suhaimi Selamat; *Burhanuddin Yeop Majlis*. Challenges in Implementing RFID Tag in a Conventional Library. Pages: 258 – 262 2006 IEEE International Conference on Semiconductor Electronics, Year: 2006.

[2]. Xuechen He.The Two-Dimensional Bar Code Application in Book Management 2010 International Conference on Web Information Systems and Mining.Volume: 1 Pages: 409 - 411, DOI: 10.1109/WISM.2010.58.

[3]. Xunrui Duan; Qingjie Zhao;Shahzda Anwar. Identifying books in the library using line segment detector and contour clustering.2012 7th International Conference on Computing and Convergence Technology (ICCCT)Pages: 998 – 1003.

[4]. K. Iwata; K. Yamamoto; M. Yasuda; K. Kato; M. Ishida; K. Murata.Book cover identification by using four directional features field for a small-scale library system. Proceedings of Sixth International Conference on Document Analysis and Recognition Year: 2001 Pages:582-586,DOI:10.1109/ICDAR.2001.953856.

[5]. Spencer G. Fowers; Dah-Jye Lee; Guangming Xiong.Improved library shelf reading using color feature matching of book-spine images.2010 11th International Conference on Control Automation Robotics & Vision Pages:2160-2165,DOI: 10.1109/I-CARCV.2010.5707391.

[6]. D. Lowe. “Object recognition from local scale-invariant features,” in ICCV, Washington DC, 1999, pp.1150-1157.

[7]. David G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints.” International Journal of Computer Vision, 2004.

[8]. Relja Arandjelović;Andrew Zisserman. Three things everyone should know to improve object retrieval.2012 IEEE Conference on Computer Vision and Pattern Recognition Year:2012 Pages:2911-2918, DOI:10.1109/CVPR.2012.6248018.

[9]. O.Chum and J. Matas. “Randomized RANSAC with Td,d test,” Image and Vision Computing, 2(10), 2004, pp.837-842.

[10]. Book Detection and Grasping in Library Scenario, Stefan Heyer; Bashar Enjarini; Christos Fragkopoulos; Axel Graeser, ROBOTIK 2012; 7th German Conference on Robotics, Year: 2012, Pages: 1 - 6