MVA2015 IAPR International Conference on Machine Vision Applications, May 18-22, 2015, Tokyo, JAPAN

Robust Visual Analysis for Planogram Compliance Problem

Anurag Saran[1]
Cadence Design Systems, India
asaran@cadence.com

Ehtesham Hassan and Avinash Kumar Maurya
TCS Research, New Delhi, India
ehtesham.hassan, avinashkumar.maurya@tcs.com

Abstract

This paper presents a novel visual analysis based framework for automated planogram compliance check in retail stores. Our framework provides an efficient and convenient solution for ensuring planogram compliance by real-time analysis of the shelf image acquired in freehand manner. We present a novel application of Hausdorff metric for occupancy computation in product shelf images. Subsequently, we present a robust solution for product counting which applies robust row detection algorithm, and exploits texture and color feature for accurate counting. In this context, our system addresses the most general scenario of multiple varieties in single product type. The empirical validation of our framework is demonstrated on range of real-life images from stores located across different geographies, where it has achieved satisfactory and encouraging results.

1 Introduction

In this paper, we present a novel visual analysis based method for evaluating planogram compliance of a product shelf. We require only the product shelf image captured on-line by a smart phone, where an ideal image describing the planogram compliant state of the product shelf and corresponding product image templates are stored in a database. Using this our framework identifies the non-compliant locations, as well as product count in an efficient and user friendly manner.

1.1 Related Works

The planogram (fig.1) is a diagram or model that indicates the placement of retail products on shelves for customer ease and convenience, resulting maximum sales. The research in retail marketing has shown that maintaining the planogram compliance can improve the overall profit by significant measure [1].

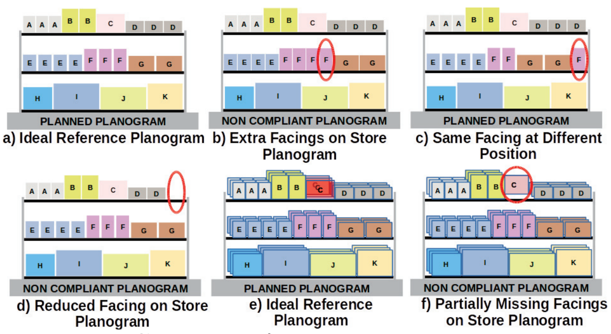

Figure 1: Compliant & non-compliant planogram.

Most of the previous approaches dealing with the automation of this problem rely on either using RFID tags [2–4], or ancient accounting and inventory management methods sometimes augmented with data from an array of sensors [5].

The RFID based approach faces the problem of exponential cost of sensor installation and time-consuming hard-work of attaching tags with every product and removing them again at the billing counter. Additionally inspite of being capital intensive, it is not able to produce desired and completely correct results [4]. The inventory based methods maintain the product log at the checkout point and work with assumption that if a product is selected, it would be replaced at the same spot. A fallacy which is shared with some of the RFID/Sensor based approaches. These assumptions cannot be considered to be valid specially in conditions of high traffic, or during the off-season when low stocks are maintained on the shelves. In this problem domain, some image analysis based methods have been proposed which can cater to some of the problems discussed above. Nevertheless, some of these approaches rely on fixed/mounted cameras for monitoring requiring template images for all of the products available in the store [6]. Also this approach is not able to properly test planogranm compliance, if the front-face of product is not clearly visible [6], [7]. Some even rely on real-time monitoring, been done by sales reps at the back-end [8].

Major Contributions: With respect to the observations discussed above, our contributions are as follows. We present an efficient framework for planogram compliance check in retail marketing scenario by a simple smart-phone click of a specific product shelf. we define an innovative application of Hausdorff metric for occupancy computation. Also, we propose a simple combination of texture, and color feature for robust product count. In particular, our framework provides the following solutions.

(1) Occupancy computation: Partial or Complete presence of different products(fig. 1).

(2) Product location: Positions and placements of products.

(3) Product count: Number of different products in the shelf.

2 Our Framework

Our framework consists of three modules: Row extraction, Occupancy computation, and Count and placement identification. The flow starts with product shelf image acquisition, this is followed by some pre-processing steps to extract the ROI (Region of Interest). The row extraction then identifies different shelf rows. Subsequently, these rows are processed for the occupancy, count and placement computation.

[1]Contributed during his employment with Tata Consultancy Services, India

Preprocessing: ROI extraction First step is the perspective distortion rectification of example image which is captured by a hand-held mobile phone camera (or any other suitable device). The user selects corner points of the product shelf using the touch screen. Using these four points, we rectify the perspective distortion in image, using the conventional projective geometry rectification [9] as shown in fig. 3.

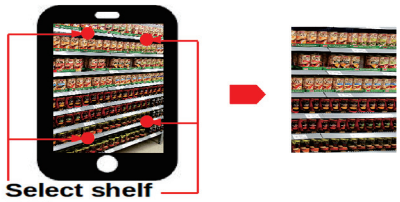

Figure 3: Corner Selection for Rectification

2.1 Occupancy computation with respect to the reference

This approach works on basis of comparison of example image describing the ROI of product shelf with a reference image. The reference image stored in the database depicts the planned planogram for the product shelf as shown in figure 1. Therefore, any deviation in the example image from this should be a identified.

In particular, we encounter following scenarios while computing the occupancy: missing products (both complete and partial i.e. empty front protion in shelf rows), misplaced products, and extraneous products(fig. 1). Broadly, they can be described by only two classes: i) Completely, and ii) Partially missing case including misplacements, extraneous products and empty front portions.

2.2 Hausdorff map based occupancy computation

The point-to-point comparison of example with reference image will not be effective because of small shifts, and misalignments due to the nature of the application. Therefore, we generate a local distance map for example and reference image based on Hausdorff metric [10]; which is subsequently applied for comparison. The metric is defined in eqn.(1).

$$H(A, B) = \max\{h(A, B), h(B, A)\} \quad (1)$$

Here A, and B are the local spatial areas of reference, and example image which are being compared. Whereas a and b (in eqn.(2)) are smaller divisions in A and B, i.e., A and B are the super-masks, and a and b denote sub-masks in corresponding super-masks. For all experiments discussed in this paper, super-masks A and B were selected of 9×9, whereas sub-masks a and b were of the size 3×3 pixels respectively. $h(A, B)$ in the eqn. 1 denotes the directed Hausdorff distance given as follows:

$$h(A, B) = \max_{a \in A}\{\min_{b \in B}\{d(a, b)\}\} \quad (2)$$

The selection of d would be discussed later in this section. Figure 2 shows the block-level representation of our occupancy computation method using Hausdorff metric based distance map. It has two separate block describing individual identification of complete and partial missing cases. We describe them separately in the following discussion.

2.2.1 Identification of completely missing case

For the identification of completely missing product case, we have applied *Euclidean distance* as the comparison metric $d(a, b)$ with all computations performed in RGB color space (as shown in figure 2). We observed that this metric in the selected color space is robust for identifying the missing, misplaced and wrongly placed products due to the prominient difference in RGB intensity values with reference. Here, the sub-masks a and b were expressed as the mean value of pixel intensities in RGB planes in the defined region of 3×3. Using these values of a, b, and d as Euclidean distance, we plot the distance map between example and reference image, and apply Otsu's thresholding [11] criterion. The foreground in the binary image would correspond to the completely missing case as shown in the upper portion of figure 2.

2.2.2 Identification of partial missing case

Referring figure 1(e), a product is considered to be partially missing when the corresponding row is par-

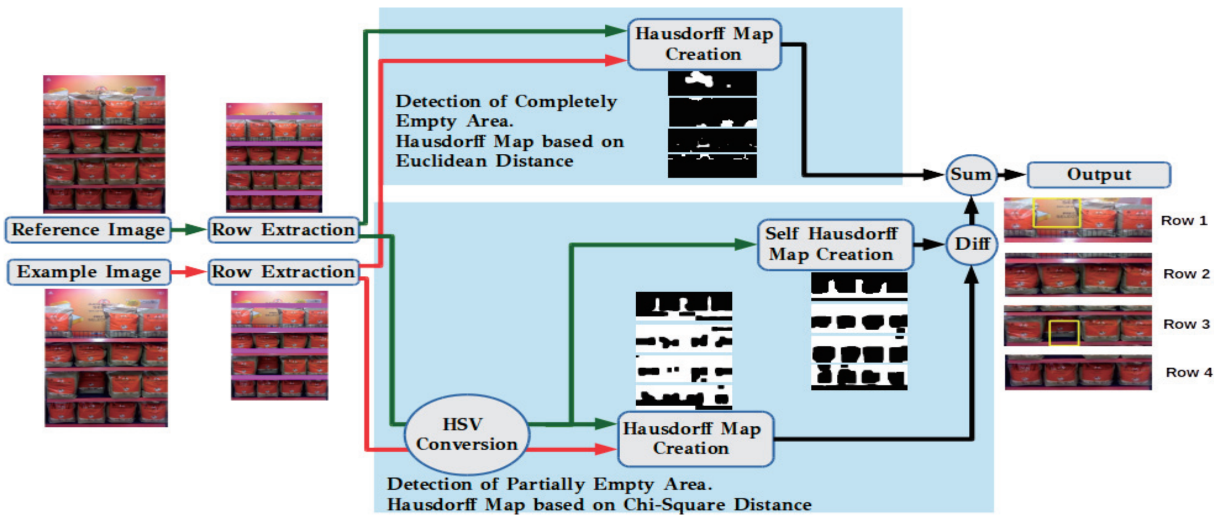

Figure 2: Comparison of Reference and Current Image for Occupancy Computation

tially depleted. This leads to an inherent shade on the packets placed at inner rows in the stack. This causes decrease in the light intensity value onto the packet; easily to be captured with the distribution of value channel in HSV space. Therefore, we define the sub-masks a and b expressed as Value (V) distribution in their defined region of 3×3. For the purpose of $d(a, b)$ computation, *Chi-Square* distance was applied as shown in equation 3

$$d(H_1, H_2) = \sum_{I} \frac{(H_1(I) - H_2(I))^2}{H_1(I)} \qquad (3)$$

Here, H_1 and H_2 are the histograms of the sub-masks of reference and example image. Nevertheless, this approach generates resulting distance map with noisy artifacts at product boundaries. We address these artifacts by generating the self Hausdorff map of the reference image using horizontal and vertical translation as shown in the figure 4. Understandably, the distance for the translation has to be the difference between size of super and sub-masks. The self Hausdorff map is then subtracted from the example image distance map, which filters out the noisy artifacts. Further, thresholding distance map with Otsu's [11] criterion, we get the binary distance map in which the foreground would correspond to partial missing cases as shown in the lower portion of figure 2.

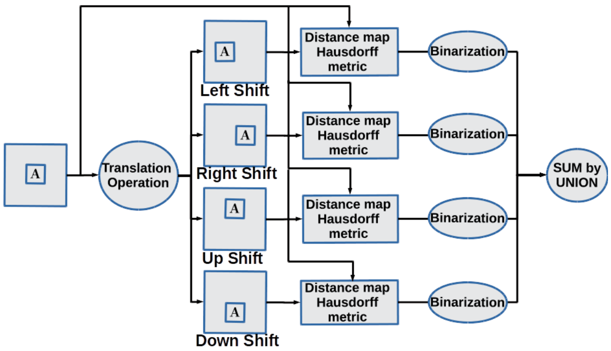

Figure 4: Self-Hausdroff map creation

Combination of the completely, and partial missing cases give the occupancy estimate for the product shelf. Furthermore, the accuracy and efficiency of this module improves if the input is in form of rows, that have been extracted by the row identification module discussed in the next section.

2.3 Product shelf row identification

In the next step, we extract different rows of product shelf. The row identification not only improves overall computing time, but also increases recognition accuracy; as the product detection is reduced to the scanning the region between two rows.

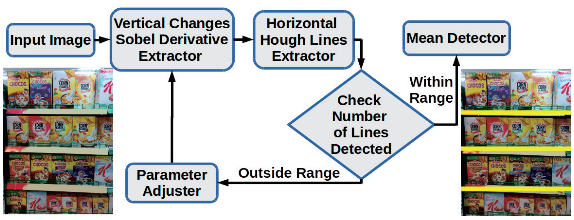

Figure 5: Row Extraction

Our approach is for extraction of rows is based on the premise that the most prominent horizontal line in the shelf will be divisions or row partitions. First, we find all vertical changes in the image using Sobel derivative giving us *chg_img*. Next, we apply Hough transform [12] on *chg_img* to detect all possible straight lines.(refer fig. 5)

An important problem in this computation is the scenario of detection of large number of uniformly spaced lines in crowded product shelves. This causes error in the accurate detection of rows. To overcome this problem, we propose an automated solution for selection of related parameters, i.e., Sobel derivative threshold, and Voting threshold in Hough transform based line detection. The solution is described below (Refer algorithm 1). Once converged, we have optimum value of these parameters for given *chg_img*.

Algorithm 1 Line Reduction Algorithm

```
procedure REDUCELINES(sobTh, hoghTh, img)
    maxLines ← (img.ht/10)
    while linDtcd > maxLines do
        linDtcd ← HOUGHLINES(sobTh, hoghTh, img)
        incr ← (linDtcd/maxLines)
        hoghTh ← (hoghTh + incr)
        sobTh ← sobTh + (incr/2)
    end while
    return sobTh, hoghTh
end procedure
```

2.4 Product Count and Placement Computation

After row detection, the analytical objective is to get product counts and their locations with respect to rows. Considering the current retail marketing, we have several very similar products having very minor differences. Our row level product count assumes the availability of exact product templates. The approach follows two steps: i) We do detection based texture properties of product image, ii) Subsequently, we eliminate false positives using color features. The details are as follows.

Detection using texture features: We have used SURF feature [13] based detection for first level of product detection. For identifying the region to be matched with templates, we follow a sliding window approach. As we have rows already detected, we need to identify the sliding window width with respect to the aspect-ratio of the product-template. The height of the window being equal to the height of the row itself. To find the sliding window width, we multiply the aspect-ratio(w/h) of the template with window height. Also, in subsequent iterations, instead of sliding the window by a pixel or by the window width, we move it by 1/3 of the window width. This is a trade-off for ensuring both detection speed and accuracy.

False positive removal using color feature The SURF based detection generates many false positives for many flavours of one product primarily differentiable only by color while keeping the packing design and graphics similar. For addressing this situation, we apply next level of filtration for removal of false positives by matching the color histogram between the template image and the detected regions in example image (refer figure 6).

Since, the region and template may be of different dimensions, we apply normalized histogram and use

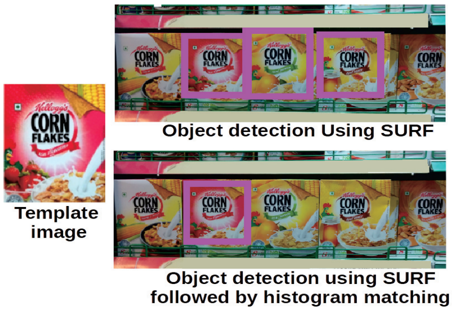

Figure 6: Detection: Row1 - SURF; Row 2 - SURF + Color Histogram

Bhattacharya distance for measuring the similarity between two distributions. The Bhattacharya distance between two equalized histograms h_1, and h_2 is defined as in eq. 4.

$$d(h_1, h_2) = \sqrt{1 - \sum_i \sqrt{h_1(i) \cdot h_2(i)}} \tag{4}$$

3 Empirical Evaluation and Analysis

We have evaluated our framework on eight collection of product shelf images from different retail stores across the world captured during different times. We also have corresponding reference images, and product template images. These images varied from 2448×3264 to 598×800. After perspective rectification, we normalize all of these images to the size of 220×300 for occupancy computation. The results related to various aspects are as follows:

Accuracy: The efficacy of our approach has been evaluated based on the subjective evaluation. The occupancy computation basically requires the verification of shelf areas having complete or partial missing cases. Table 1 summarizes complete results on all images. For occupancy detection, where the objective was to detect the planogram non-compliant points(violations), our approach has correctly identified all such cases. Similarly, for row detection, we were able to detect all rows in our images collection. For product counting, we evaluated on 25 different types of different sizes covering $25 \sim 75\%$ of shelf height. Nevertheless, our solution achieved perfect result for only 23 product types as the two true negatives were were soft plastic bags having irreglar shapes in vertical placements. Here the color histogram macthing was done with 20 bins for each plane in RGB sapace.

Table 1: Complete results

Module	# of violations/rows/products	# of our detections
Occupancy det.	36	36
Row extraction	92	92
Product ident.	25	23

Efficiency and Robustness: Figure 7 shows screen-shot of our planogram compliance evaluation framework. Our approach is highly efficient. Our sub-optimal implementation on an ordinary 2GB RAM, i3 processor based workstation takes less than 1 second process one image of all sizes available in the collection. Our approach needs to scan the rows of the shelf only in one dimension, instead of scanning the whole shelf along the height and width. It reduces the application run time complexity from $O(n^2)$ to $O(n)$. Also the robustness is ensured in the occupancy computation stage by employing self Hausdorff map based filtration. Also, the techniques used for row extraction and placement detection are translation invariant.

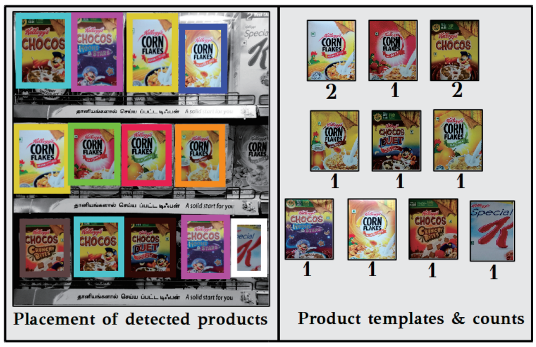

Figure 7: Product Count and Placement Workbench

4 Conclusion

Our approach for ensuring planogram compliance is robust efficient and user friendly. Also unlike other related approaches, occupancy computation module is able to detect all the differences from the ideal condition, irrespective of the product face being visible or not. However the product placement detection module suffers from this limitation like other related approaches. Therefore further progress can be made in not only improving this module but also in figuring out an approach such that even the product templates are extracted automatically from the reference image.

References

[1] W. Bishop: “Documenting the value of merchandising.” National Assc. for Retail Merchandising, 2000.
[2] Chaves, L.W.F.: “Planogram compliance using automated item-tracking,” US Patent 12/548,480, 2011
[3] Noonan, W.: “System and method of updating planogram information using RFID tags and personal shopping device,” US Patent 7,493,336, 2009
[4] F. Chaves et all: “Finding Misplaced Items in Retail by Clustering RFID Data”, *Conference on Extending Database Technology*, pub. ACM, pp.501–512, 2010
[5] E. Frontoni et all : “Information Management for Intelligent Retail Environment: The Shelf Detector System,” *Information*, vol.5, no.2, pp.255–271, 2014
[6] Opalach, A. and Fano, A. and Linaker, F. and Groenevelt, R.B.: “Planogram extraction based on image processing” US Patent 8,189,855, 2012
[7] Visual Shelf Monitoring: http://mathecsys.com/visual-shelf-monitoring
[8] TRAX Retail Solns.: http://traxretail.com/why-trax/
[9] Forsyth, David A. and Ponce, Jean: “Computer Vision: A Modern Approach” pub. Prentice Hall, 2012
[10] Huttenlocher et all: “Comparing images using the Hausdorff distance”, *Pattern Analysis and Machine Intelligence* vol.15, pg.850–863, 1993
[11] N. Otsu, Threshold selec. method from gray level hist., IEEE Trans. Syst. Man Cybern.,pp. 6266 1979.
[12] Varol G. et all: “Product placement detection based on image processing”, *Signal Processing and Communications Applications Conference* pg.1031-1034, 2014
[13] Bay et all: “Surf: Speeded up robust features” *Computer Vision–ECCV* pub.Springer, pg.404–417,2006
[14] A. Bhattacharyya: Divergence between two statistical populations. Bull. Calcutta Math. Soc. 55(1943)