First online: 30 August 2016

Fingerprints for Imposed Layers in Document Images Based on Huffman Code and Logical Layout Analysis

Surabhi Narayan[1], Sahana D Gowda[2]

[1,2]Department of Computer Science & Engineering, B.N.M Institute of Technology, Bengaluru, India

[1]surabhi.narayan@gmail.com; [2]sahanagowdad@gmail.com

***Abstract*-A document is characterized by its layout and component structure. Document layout is due to the placement of the content components and document structure is due to the geometrical shape of the content components. Content components in a filled-in document image consist of general information foreground layer and vital information imposed layer. The foreground layer consists of printed text, logos, tables and lines that are identical for documents of the same class; the imposed layer of the document image consists of handwritten text, signatures and seals imposed on the document image that are unique to every document image. Processing filled-in document images for indexing, considering general information along with vital information is complex with the possibility of generating identical indexes due to large amount of general information suppressing fewer imposed layer vital information. In this paper, a novel technique was proposed to generate a unique code by formulating a logical layout of the imposed layer which was extracted from the filled-in document image using registration. The extracted imposed layer components were represented by centroids based on their spatial occupancy and the imposed layer was hierarchically decomposed into 16 equal quadrants. The Huffman tree generation algorithm was applied based on the number of centroids in a quadrant and with quadrant indices were assimilated to generate a unique code for the logical layout of the document image. In order to verify the applicability of this method, extensive experimentation were conducted on extracted imposed layers from application forms, student records, bank cheques and declaration forms.**

Keywords- Imposed Layer; Centroids; Huffman Codes; Quad Decomposition; Logical Layout

I. INTRODUCTION

Filled-in document images consist of general information foreground imposed on a null background and vital information layer imposed on general information foreground. Printed text, logo, and tables lines are identical to a class of documents constituting the foreground layer. Printed/handwritten text, signature and seal that are unique to every document image constitute the imposed layer. Filled-in document images with identical general information but different vital information tend to produce identical indexes due to large amount of general information masking vital information. Therefore the imposed layer was extracted from the document image by document image registration. After extraction, the imposed layer did not have a physical/logical layout structure to define the visual appearance of the imposed layer. Hence logical layout of the document image was formulated based on the spatial occupancy and adjacency relationship among imposed layer components which were represented by centroids. The imposed layer was decomposed by quad decomposition at two hierarchical levels. The position of centroids in the quadrants depicted the spatial occupancy. The components at Level 1 hierarchy in quad decomposition specified adjacency at a superficial level. To get an in-depth analysis of adjacency, Level-2 decomposition was applied to further decompose every quadrant of Level 1 thereby generating 16 quadrants. Quadrants were indexed in counter-clockwise direction from Level 1 to Level 2. Quadrant indices with number of centroids were used to construct Huffman tree which depicted a logical layout for the imposed layer. The leaf nodes were the quadrants with the number of components in them. Subsequent merge in the tree construction generated internal nodes. The complete tree with root node indicated the logical summation of quadrants of the imposed layer. Every logical layout was represented by a code to quantify the uniqueness in the spatial occurrence and the adjacency. Spatial variation in the position of centroids and adjacency varied the logical structure of the imposed layer. Two levels of hierarchical quadrants were indexed by four bit binary codes and Huffman codes were generated for every quadrant with centroids. Quadrant codes and Huffman codes were correspondingly assimilated in counter-clockwise direction of occurrence in Level-2 hierarchical decomposition in bottom to top fashion. The codes were of variable length depending on the occurrence of components, spatial occupancy and adjacency. The proposed model was justified by extensive experimentation.

II. STATE OF ART

Document image has a unique appearance due to the structure and placement of content components. Document Layout analysis is the process of representing the document image in terms of placement and relationship among the content components [1]. Geometrical layout analysis and logical layout analysis are the two approaches to document layout analysis [2, 3]. Geometrical layout analysis aims to produce a description of the geometrical structure of the document image. Geometrical

layout analysis represents the content components at various hierarchical levels of details like text, graphics, page, paragraph, line or word. Logical layout analysis assigns meaningful labels for the homogenous regions and produces a description for the logical relationship among the homogenous regions.

Asada et al represented logical layout of a document image in the form of a tree [4]. In this method, a physical tree is constructed from the content components and is transformed into a logical tree. Logical labels are assigned to content components using rule based grammar. Fisher et al proposed a rule based model to recognize geometrical layout and transforming it into a logical layout structure [5]. Location, format and textual cues aid in rule building. The method determines the reading order of text blocks and expresses it in terms of document mark-up language. Conway et al proposed page grammars and page parsing technique to recognize the structure of the logical layout [6]. Grammar rules are defined based on the neighbour relationship like left of, over, left-side, close-to etc.

Logical layout analysis techniques available in literature use a priori knowledge to determine the reading order and hence are supervised techniques. Uniformity and positional cues are used to assign labels and define relationship among them. Extracted imposed layer of a document image are highly non-uniform in nature in terms of structure and spatial occurrence. In an unsupervised environment, positional cues or reading order is not available for the components in the imposed layer.

Works are also reported to generate representation for document layout by eliminating text and retaining lines, tables and line crossings in the document image. These techniques generate layout representation by analyzing lines, grid structure and component blocks [7-9].

Pirlo et al proposed document image layout analysis based on Radon transform for retrieval of invoices, waybills and receipts [7]. The considered document images are defined by a grid structure. The text components are removed and the grid is extracted from the document image. Radon transform is used to extract the feature vector and matching is performed using dynamic time warping. Duygulu et al proposed a logical representation of form documents for identification and retrieval [9]. Thick lines and thin lines are extracted from the document image and used to form blocks. A heuristic approach is used to group blocks with similar information using the length and thickness of lines. Blocks are inserted into a tree to represent the logical structure. Yoshitake et al proposed a document layout structure generator based on component blocks [10]. Component blocks are generated using projection profile and blocks are inserted as nodes into trees. Neighbouring text block nodes are combined in a hierarchical tree building process. Cesarini et al proposed retrieval of document images based on layout similarity [11]. The document image layout is described by means of a Modified XY tree. Each page of the document is represented by a feature vector based on global features of the page. The global features describe the position and size of the printed part of the page. Shin et al proposed document image retrieval based on layout similarity [12]. The similarity measure considers spatial structure computed by aggregating the content area. The structural similarity is measured by computing area overlaps. Percentage of overlap for each region is computed and summed. The documents are ranked according to the overall summed percentage. Gao et al proposed document image matching by constructing dendogram that defines the structural relationship among regions [13]. SIFT descriptors and hierarchical K-means clustering are used to generate labels for every region. Key regions in terms of assigned labels and their bounding boxes are stored in the database for matching.Hu et al proposed document image classification using layout analysis [14]. The structural characteristics of regions are encoded using fixed length vectors and a hidden markov model based page layout classification system is designed. Punitha et al proposed indexing and retrieval of document images by spatial reasoning [15]. The connected component technique is applied to identify the component blocks and nine directional spatial relationships are defined by constructing a B-tree.

Logical layouts for document images are formulated by transforming the geometrical layout and by eliminating the text and analysing the grid structure. The imposed layer after extraction loses geometrical and logical layout because foreground information is eliminated. No work has been reported in literature that formulates logical layout of processed document images containing only imposed layer components. Logical layout analysis of document images with only imposed layer components poses challenges as elimination of general information causes the image to lose its layout structure. In this paper a novel code generation technique was proposed by formulating the logical layout by constructing a Huffman tree based on the spatial occupancy and adjacency relationship among components in the imposed layer.

III. PROPOSED MODEL

Imposed layer components are placed at pre-defined location based on the layout structure of the foreground general information. The imposed layer components were extracted from the filled-in document image using document image registration based on geometric invariance and Hausdorff distance [16]. Geometric invariant points were identified on the template and filled-in document image and transformation values were computed. The filled-in document image is aligned to the template and imposed layer components are extracted. Extracted imposed layer lost its layout structure due to the removal of foreground. Components in imposed layer were free format; did not have fixed size/structure as they were handwritten. Components are sparsely placed where generating geometric co-relation or logical reading order is not possible. The spatial occupancy of these components spread across the region of interest non-uniformly. Therefore the components were represented by centroids with dimensional computation. Components in the imposed layer were irregular in shape and consisted of multiple connected components. Therefore morphological dilation was applied to connect the neighbouring pixels

within the component. Morphological dilation is the process of adding a layer of pixels to the outer and inner boundaries of regions to merge the neighbouring sub components within a component [17]. Centroids were computed for every dilated component in the imposed layer to define the spatial occupancy.

A. Centroid Computation

The Centroid of a component was computed in terms of x and y co-ordinates. Centroid is the center of mass of uniform density of a geometrical object. Centroid of geometrical object of irregular shape is defined as follows [18, 19]:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i * A_i}{\sum_{i=1}^{n} A_i}, \bar{y} = \frac{\sum_{i=1}^{n} y_i * A_i}{\sum_{i=1}^{n} A_i} \quad (1)$$

where x_i and y_i is the distance to the split region centroid and A_i is the area of the split region. The irregular geometrical shape is split into multiple(n) regular shaped regions.

Due to the sparse unstructured non uniform spread of centroids, spatial occupancy and adjacency in terms of geometry was not possible. So the imposed layer is decomposed using quad decomposition at two levels of hierarchy [20].

Quad decomposition of Q was defined orthogonally to the axes according to the following:

$Q2 = \left((x_3, y_3)\ ,\ (x_4, y_4)\right)$	$Q1 = \left((x_1, y_1)\ ,\ (x_2, y_2)\right)$
$Q3 = \left((x_5, y_5)\ ,\ (x_6, y_6)\right)$	$Q4 = \left((x_4, y_4)\ ,\ (x_7, y_7)\right)$

where the following were true:

$x_1 = \text{Min}(x)\ and\ y_1 = \frac{\text{Max}(y) - Min(y)}{2}$	$x_2 = \frac{\text{Max}(x) - Min(x)}{2} and\ \ y_2 = \text{Max}(y)$
$x_3 = \text{Min}(x)\ and\ y_3 = \text{Min}(y)$	$x_4 = \frac{\text{Max}(x) - Min(x)}{2} and$ $y_4 = \frac{\text{Max}(y) - Min(y)}{2}$
$x_5 = \frac{\text{Max}(x) - Min(x)}{2} and\ y_5 = \text{Min}(y)$	$x_6 = \text{Max}(x)\ and\ y_6 = \frac{\text{Max}(y) - Min(y)}{2}$
$x_7 = \text{Max}(x)\ and\ y_7 = \text{Max}(y)$	

In Level-1 hierarchy, imposed layer was divided into four quadrants. At this level, partial placement of components at sub component/neighbouring positions causes components to be closer. Their adjacency measure is lost as they fall into the same quadrant. To overcome this Level-2 hierarchy decomposition was applied to further divide each Level-1 quadrant into four sub-quadrants. Every first-level quadrant had four sub-quadrants which accomplished an inner distance separation among the components in one quadrant of level-1 hierarchy. The decomposition and numbering strategy is depicted in the Fig. 1.

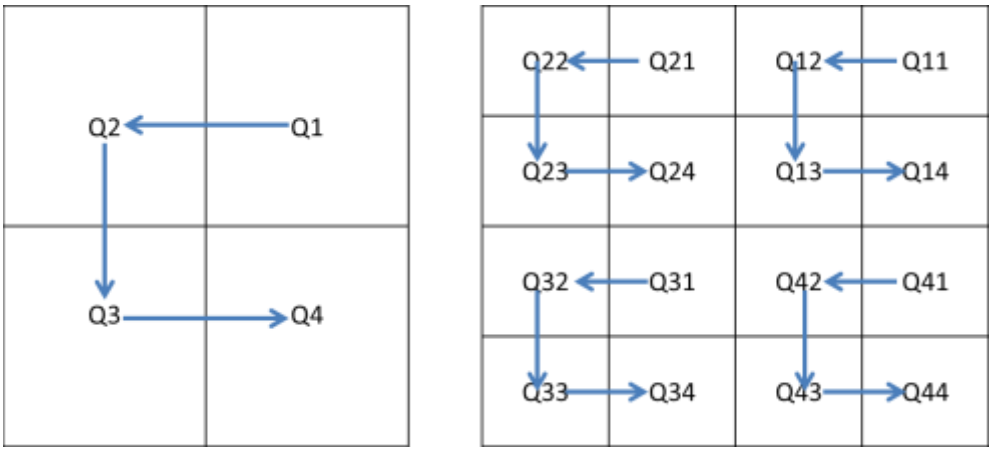


Fig. 1 Quad decomposition at level-1 and level-2

The quadrants were indexed in Level-1 hierarchy in counter-clockwise direction. Every indexed quadrant entered into indexing Level-2 hierarchy in counter-clockwise direction. Every quadrant was indexed by two numbers which are represented by four binary bits as shown in the Fig. 2a and 2b.

Q22	Q21	Q12	Q11
Q23	Q24	Q13	Q14
Q32	Q31	Q42	Q41
Q33	Q34	Q43	Q44

Fig. 2a Quadrant indices

0101	0100	0001	0000
0110	0111	0010	0011
1001	1000	1101	1100
1010	1011	1110	1111

Fig. 2b 4 bit binary codes

The imposed layer with two-level decomposition was divided into 16 quadrants. The centroids of the components in the quadrants depicted the spatial occupancy and centroids neighbouring in adjacent quadrants implied adjacency amongst them. To implicate adjacency relationship, quadrants were set as leaf nodes with number of centroids in the node for the construction of Huffman tree [21].

A Huffman tree is a full binary tree in which each leaf node corresponds to a symbol in the information content. A Huffman tree is constructed based on the frequency of the occurrence of symbols. The traversal of Huffman tree generates Huffman codes which are widely used in lossless data compression and transmission. A Huffman code is a variable length code generated for every leaf node of the Huffman tree by traversing the tree from root to leaf. Symbols with higher frequency are encoded with shorter code-words and symbols with lower frequency are encoded with longer code-words.

B. Huffman Tree Generation

Quadrants of Level 2 hierarchy may contain zero or more centroids based on the positional occurrence of the imposed layer components. The number of centroids and quadrant index were used to build the logical adjacency among the imposed layer components. Quadrant indices with non-zero centroids were considered for Huffman tree construction. Centroids together with quadrant indices formed the leaf nodes of the Huffman tree. The tree was constructed by recursively merging two nodes that enclosed the least centroids. Tree construction was complete when all nodes were merged. Traversal of Huffman tree from root to leaf generated unique code for every quadrant defined using Level-2 hierarchy.

To derive logical analysis with a layout structure, Huffman tree with tree traversal was implemented as described below.

Huffman tree generation algorithm:

i. Initialize the quad sequence wherein every element contains number of centroids along with the quadrant index.

ii. Create a leaf node for every element in the quad sequence. Leaf node contains the number of centroids in the quadrant along with the quadrant index.

iii. Sort the quad sequence in increasing order of the number of centroids. Extract the least two elements. Identify the nodes corresponding to the extracted elements. Replace the two elements in the quad sequence by the sum of number of centroids of the extracted elements.

iv. Merge the identified nodes to generate an internal node. Internal node contains the sum of the number of centroids of its child nodes. Repeat steps(ii) and (iii) until all leaf nodes are exhausted/ connected to the tree.

v. Assign binary code 0 to the left links of the tree.

vi. Assign binary code 1 to the right links of the tree.

A hypothetical Huffman tree is depicted in the Fig. 3. Let quadrants Q_{ab}, Q_{cd}, Q_{ef} and Q_{gh} contain non-zero centroids. Quadrant Q_{ab} contains A number of centroids, Quadrant Q_{cd} contains B number of centroids, Quadrant Q_{ef} contains C number of centroids and Quadrant Q_{gh} contains D number of centroids. It was assumed that $A \leq B \leq C \leq D$.

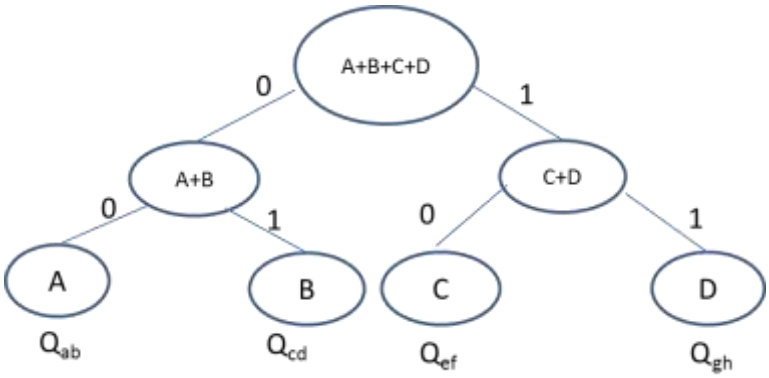


Fig. 3 Hypothetical Huffman Tree

Traversal of the Huffman tree from root to the leaf generated a prefix code/Huffman code for every leaf node. The generated code was a variable length code. Quadrants enclosing more centroids were encoded with fewer bits whereas quadrants enclosing a lesser number of centroids were encoded with more number of bits.

For the Huffman tree shown in Fig. 3, the codes were as follows:

Q_{ab} – 00

Q_{cd} – 01

Q_{ef} – 10

Q_{gh} – 11

Quadrant indices and Huffman code were assimilated correspondingly to generate a unique code that depicted the logical layout of the imposed layer. Experimental analysis for various types of document images is presented in the next section.

IV. EXPERIMENTATION

Filled-in documents with different layouts were considered to test the efficacy of the proposed model. The imposed layer from the filled-in documents was extracted using document image registration.

A. Data Sets

Three types of document images with different imposed layer layout were considered to test the efficacy of the proposed model. Application form images contained a null background, general information foreground, and vital information imposed layer. The content components in the imposed layer were densely spread across the image layout. Bank cheques contained water marked background, general information foreground, and vital information imposed layer. The imposed layer contained fewer content components. Student enrollment form images contained a null background, general information foreground and vital information imposed layer. Content components were densely spread across the layout. The data set consisted of 300 application form images, 100 cheque images and 300 student enrolment form images. All images in the data set consisted of printed text, handwritten text and signatures.

B. Application Form

The application form was composed of null background, general information foreground and vital information imposed layer. Foreground layer consisted of printed text and logo, the imposed layer consisted of handwritten text, signature and seal components. The imposed layer from the application form was extracted using document image registration based on geometric invariance and Hausdorff distance [16].

Fig. 4 shows the document image with null background, general information foreground and vital information imposed layer. Components of the imposed layer were distributed throughout the document image. Fig. 5 represents the imposed layer extracted using document image registration [16]. The extracted imposed layer was recursively decomposed into 16 quadrants and the quadrant indices were assigned 4 bit binary codes as defined in the Fig. 2. Fig. 6 shows the imposed layer decomposed hierarchically into 16 equal quadrants. Centroids of the imposed layer components were computed using Eq. (1). Fig. 7 shows the centroids and number of centroids in the quadrants which were indexed from Q11 to Q44.

Quadrants with non-zero centroids were considered for Huffman tree generation. A quad sequence containing the number of centroids along with the quadrant index was generated. As seen from Fig. 7 Quadrants Q11, Q12, Q13, Q21, Q23, Q32, Q34, Q41 and Q43 did not enclose any centroids and hence were not considered for tree construction.

Fig. 4 Application form

Fig. 5 Imposed layer composition of application form image

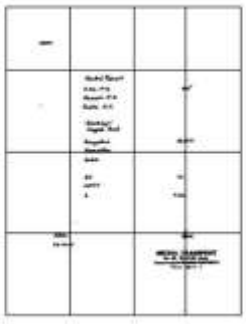
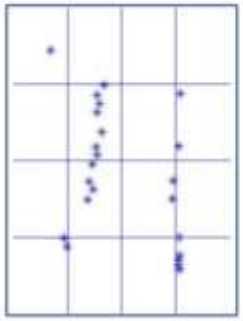


Fig. 6 Imposed layer components in quadrants



1	0	0	0
0	7	0	2
0	4	2	0
2	0	0	5

Fig. 7 Centroids in quadrants

As shown in Fig. 8, quadrants that enclosed at least one centroid were considered for Huffman tree construction. Quad sequence consisted of the quadrant index and the number centroids in that quadrant. Elements of the quad sequence formed leaf nodes of the Huffman tree. Nodes were merged recursively and sum of node elements were computed to generate internal nodes. Quad sequence was sorted at all times during the tree construction. The Huffman tree generated for the application form image in Fig. 4 is shown in Fig. 9.

Quadrant number	**Q22**	**Q14**	**Q33**	**Q42**	**Q31**	**Q44**	**Q24**
Number of centroids in the quadrant	**1**	**2**	**2**	**2**	**4**	**5**	**7**

Fig. 8 Quad sequence

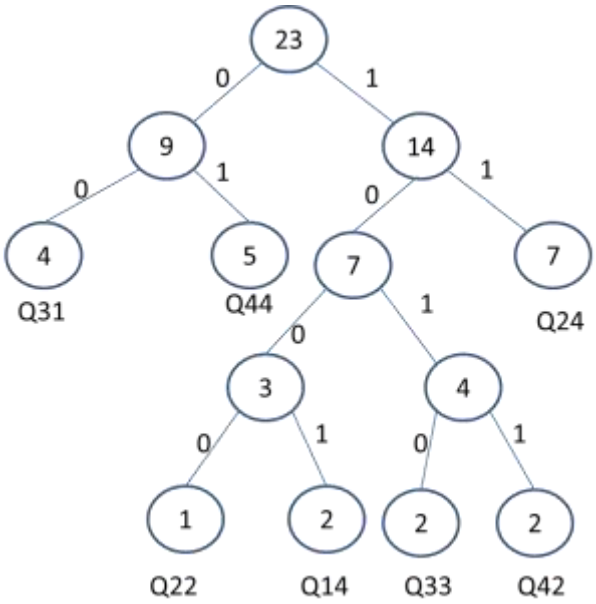


Fig. 9 Huffman tree for application form image

Traversing the tree from root to the node generated Huffman code. The generated code was a variable length code wherein, quadrants with large number of centroids were encoded with fewer bits and vice-versa.

Huffman codes:

Q14: 1001

Q22: 1000

Q24: 11

Q31: 00

Q33: 1010

Q42: 1011

Q44: 01

Quadrant indices were assigned unique four bit binary code in counter-clockwise direction as defined in Fig. 2. The quadrant code was assimilated with the Huffman code to generate a unique code for the logical layout of the document image. Fig. 10 shows the assimilation of quadrant codes and Huffman codes and Fig. 11 depicts the code for the logical layout of the application form image.

Quad code	**Huffman Code**	**Quad Code**	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**
0011	1001	0101	1000	0111	11	1000	00	1010	1010	1101	1011	1111	01

Fig. 10 Assimilation of quadrant codes and Huffman codes

Code depicting the logical layout of the application form image is shown below:

Code for the logical layout	00111001010110000111111000001010101011011011111101

Fig. 11 Code for the logical layout of application form image

Code depicting the logical layout was the concatenation of the quadrant code and the Huffman code. The generated code was a variable length code as nodes of the Huffman tree varied with the imposed layer components.

Codes generated were identical if the images had the same logical layout, i.e., the imposed layer components had the same spatial occupancy and adjacency relationship. Fig. 12 shows three extracted imposed layers with corresponding logical layout codes.

Layout code:
0010010001001100011101100011110110111 0000

Layout code:
0001001001001101010011101011110100011110100111 00101

Layout code:
0001010001001100011111100010110101111 0000

Fig. 12 Document Images with corresponding codes

C. Bank Cheque

Bank cheques are documents maintained in banks to keep track of financial transactions. Most cheques images have a similar layout to the one shown in Fig. 13. The name and logo of the bank is on the top, date is on the top right corner, name of the payee and amount are in the central part of the cheque, signature is at the bottom right and cheque number and account number are at the bottom. The imposed layer of bank cheque image is composed of name of the payee, amount, and date in handwritten form, signature, printed cheques number, and imposed account number. The imposed layer was extracted from the filled-in cheque using document image registration. The extracted imposed layer is shown in Fig. 14. The imposed layer layout was decomposed hierarchically into 16 equal quadrants. Centroids were computed for the imposed layer components using Eq. (1). Spatial occupancy and adjacency relationship is depicted in Fig. 15. The data set consisted of 100 bank cheque images.

Fig. 13 Bank Cheque

Fig. 14 Imposed layer of bank cheque

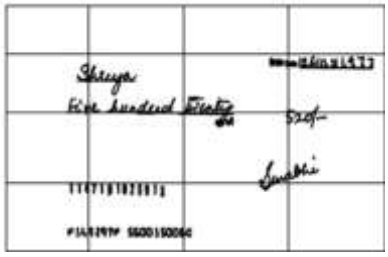
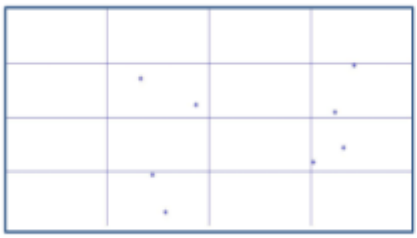


Fig. 15 Imposed layer decompostion into 16 quadrants



0	0	0	0
0	2	0	2
0	0	0	2
0	2	0	0

Fig. 16 Centroids in 16 quadrants

As can be seen from Fig. 15, imposed layer of cheque images hadfewer components as compared to the application forms.

Quadrant number	Q14	Q24	Q34	Q41
Number of centroids in the quadrant	2	2	2	2

Fig. 17 Quad sequence

As is evident from Fig. 16, only four of the 16 quadrants enclose component centroids. For the considered cheque image, all quadrants contained an equal number of imposed layer components. The quad sequence and Huffman tree are shown in Fig.

17 and in Fig. 18. Assimilation of quadrant codes and Huffman codes are shown in Fig. 19.

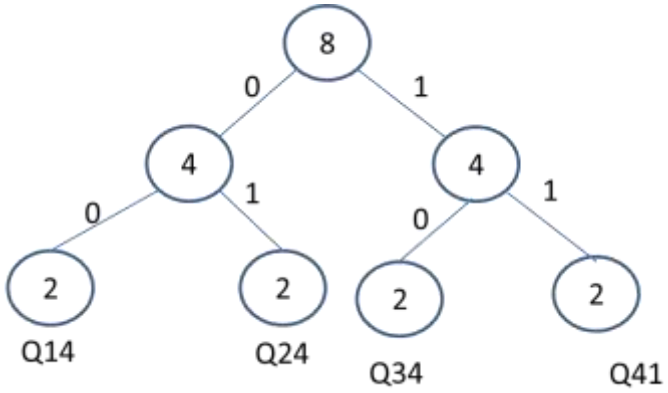


Fig. 18 Huffman tree for cheque image

Huffman codes:

Q14: 00

Q24: 01

Q34: 10

Q41: 11

Quad code	Huffman code	Quad code	Huffman code	Quad code	Huffman code	Quad code	Huffman code
0011	00	0111	01	1011	10	1100	11

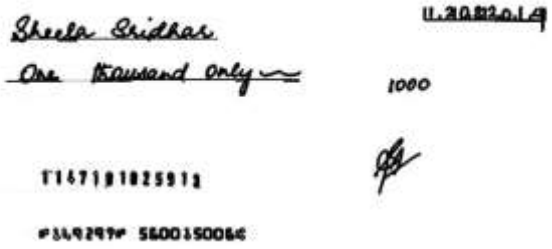
Fig. 19 Assimilation of quadrant codes and Huffman codes

Fig. 20 shows the assimilated code depicting the logical layout of the cheques image.

Code for the logical layout	00110001110110111010011

Fig. 20 Code for the logical layout of the cheque image

Layout code: 00000100011110011111101101110000



Layout code: 00111101111010110111000 0

Fig. 21 Cheque images with corresponding codes

Fig. 21 shows imposed layer layout with corresponding logical layout codes. As is evident from the image, spatial occupancy of the two images was different and hence generated two different layout codes.

D. Student Enrollment Form

Student enrollment forms are maintained by educational institutions to keep track of students' academic progress. The form contains large amounts of printed text, table, and logo. The imposed layer is composed of handwritten text and signature. Handwritten text is densely distributed in the document image. Imposed layer was extracted from the filled-in document image using document image registration. The extracted imposed layer layout was hierarchically decomposed into 16 equal quadrants. A sample student enrollment form image is shown in Fig. 22 and the extracted imposed layer is shown in Fig. 23.

Fig. 22 Student Enrollment Form

Fig. 23 Imposed layer of Student Enrollment Form

Extracted imposed layer components were analysed for spatial occupancy. Component centroids were computed based on Eq. (1). The imposed layer layout was hierarchically decomposed into 16 equal quadrants. The adjacency relationship among the component centroids is depicted in Fig. 24.

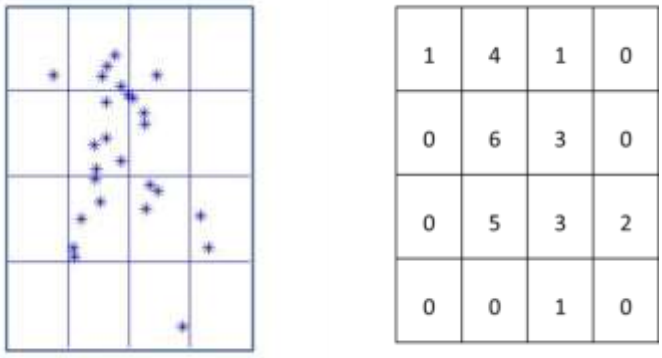


Fig. 24 Centroids in quadrants

Quad sequence showing the quadrant index and the number centroids contained in the quadrant are shown in Fig. 25. Quadrants that enclose non-zero number of centroids were considered for Huffman tree construction. The quad sequence was sorted based on the number of centroids. Elements of the quad sequence formed the leaf nodes of the tree. Huffman tree was constructed by recursively merging the two nodes with least number of centroids. The constructed Huffman tree is shown in Fig. 26.

Quadrant number	**Q12**	**Q13**	**Q21**	**Q32**	**Q24**	**Q31**	**Q41**	**Q42**	**Q43**
Number of centroids in the quadrant	**1**	**3**	**4**	**1**	**6**	**5**	**3**	**2**	**1**

Fig. 25 Quad sequence

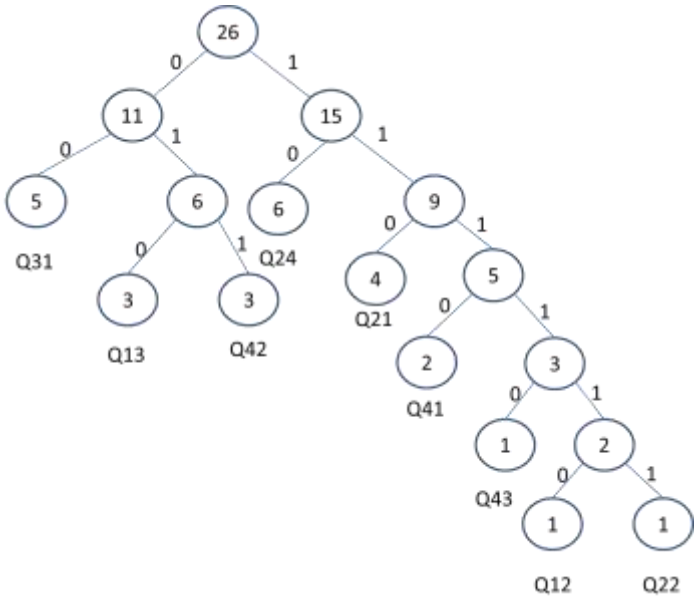


Fig. 26 Huffman tree for student enrollment form image

Huffman codes:

Q12: 111110

Q13: 011

Q21: 110

Q22: 111111

Q24: 10

Q31: 00

Q41: 1110

Q42: 011

Q43: 11110

Assimilation of quad code and Huffman code is shown in Fig. 27:

Quad code	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**
0001	111110	0010	011	0010	110	0101	111111	0111	10
Quad code	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**	**Quad code**	**Huffman code**		
1000	00	1100	1110	1101	011	1110	11110		

Fig. 27 Assimilation of quad code and Huffman code

Assimilated code for logical layout of the enrollment form image is shown in Fig. 28:

Code for the logical layout	00011111100010011001011001011111110 1 11101000001100111011010111111011110

Fig. 28 code for the logical layout of the enrollment image

V. COMPARATIVE ANALYSIS

Works are reported in literature to represent logical layout and geometrical layout of a document image. Logical layout for the document image is derived from geometrical layout. Techniques proposed in literature apply projection profile to define component blocks and based on the spatial occupancy of the component blocks, geometrical layout is defined. Logical layout is constructed based on rule grammar or visual key as the main idea is to build a hierarchy of components hence are supervised

[5, 6, 7]. Also techniques proposed in literature represent logical layout for document images with printed text only. In an unsupervised environment, positional cues are not available to label the content components. Imposed layer after extraction loses its geometrical layout structure. The proposed technique generated codes for the logical layout without any supervised knowledge or positional cues.

The proposed technique has the potential for many applications. The generated code can be used as an index to the document image database. If a unique code can be generated for the imposed layer components, the layout code in combination with the component structure code can be used as a fingerprint of a document image.

VI. CONCLUSIONS

The extracted imposed layer of a document image does not have geometrical/logical layout due to the removal of general information present in the foreground. Components in the imposed layer are free format; and do not have fixed size/structure because they are handwritten. Components are sparsely placed and hence defining geometric co-relation or logical reading order is not possible. In this paper, a novel technique to generate code for the logical layout of the document image was formulated based on the spatial occupancy and adjacency relationship among components. Imposed layer components were represented by centroids. To quantify the adjacency relationship among components, the imposed layer was decomposed into 16 quadrants using quad decomposition. Quadrants were indexed in counter-clockwise direction from Level-1 to Level-2. The number of centroids in each quadrant along with quadrant index was used to construct Huffman tree. Traversal of the Huffman tree generated Huffman code. Assimilation of the corresponding quadrant code and Huffman code generated a unique code for the logical layout of the imposed layer. Experimental results justified the correctness of the method.

ACKNOWLEDGMENT

This work is supported by Visveswaraya Technological University (VTU) under the Research Grant Scheme 2010-11 (Ref. No VTU/Aca. /2011-12 / A-9/13097). The authors acknowledge the support provided by VTU.

REFERENCES

[1] Anoop M. Namboodiri and Anil K. Jain, "Document Structure and Layout Analysis," *Digital document Processing: Major Directions and Recent Advances, Springer-Verilag, Advances in Pattern Recognition*, pp. 29-48, 2007.

[2] R. Cattoni and T. Coianiz, "Geometric Layout Analysis Techniques for Document Image Understanding: A Review," *Technical Report, IRST*, pp. 1-68, Trento, Italy, 1998.

[3] Joost van Beusekom, Daniel Keysers, Faisal Shafait, and Thomas M Breuel, "Distance Measures for Layout-Based Document Image Retrieval," *DIAL, IEEE*, vol. 30(11), pp. 232-242, 2006.

[4] S. Tsujimoto and H. Asada, "Understanding multi-articled documents," in *Proceedings of International Conference on Pattern Recognition*, pp. 551-556 (Atlantic City, NJ), June 1990.

[5] J. L. Fisher, "Logical structure descriptions of segmented document images," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 302-310 (Saint-Malo, France), September 1991.

[6] A. Conway, "Page grammars and page parsing: A syntatic approach to document layout recognition," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 761-764 (Tsukuba Science City, Japan), October 1993.

[7] G. Pirlo, M. Chimienti, M. Dassisti, D. Impedovo, and A. Galiano, "A Layout-Analysis Based System for Document Image Retrieval," *Mondo Digitale*, vol. 13(49), pp. 1-16, 2014.

[8] R. Safari, N. Narasimhamurthi, M. Shridhar, and M. Ahmadi, "Document Registration Using Projective Geometry," *IEEE Trans. on Image Processing*, vol. 6(9), pp. 1337-1341, 1997.

[9] Pinar Duygulu and Volkan Atalay, "A Hierarchical Representation of Form Documents for Identification and Retrieval," *IJDAR*, vol. 5, iss. 1, pp. 17-27, November 2002.

[10] Yoshitake Tsuji, Hiroyuki Kami, Masaaki Mizumo, Toshiyuki Tanaka, Haruhiko Tanaka, Masao Iwashita, and Tsutomu Temma, "Document Recognition System With Layout Structure Generator," *IAPR*, pp. 479-482, 1990, Tokyo.

[11] Francesca Cesarini, Simone Marinai, and Giovanni Soda, "Retrieval by Layout Similarity of Documents Represented with MXY Trees," *LNCS, DAS*, vol. 2423, pp. 353-364, 2002.

[12] Christian Shin and David Doermann, "Document Image Retrieval Based on Layout Structural Similarity," *DAS*, vol. 2, pp. 606-612, 2006.

[13] Hongxing Gao, Mar_cal Rusinol, Dimosthenis Karatzas, and Jand osep Llados, "Fast Structural Matching for Document Image Retrieval through Spatial Databases," *ICPR*, vol. 9021, pp. 939-943, 2013.

[14] Jianying Hu, Ramanujan Kashi, and Gordon Wilfong, "Document Classification using Layout Analysis," *Database and Expert System Applications*, vol. 6, pp. 556-560, 1999.

[15] P. Punitha, Naveen, and D.S. Guru, "Indexing and Retrieval of Document Images by Spatial Reasoning," *ICDCIT, LNCS*, vol. 4317, pp. 457-464, 2006.

[16] Chao Sun and Ronghai Cai, "Document Image Registration Using Geometric Invariance and Hausdorff Distance," *First International Workshop on Education Technology and Computer Science*, vol. 2, pp. 725-728, 2009.

[17] Mariusz Jankowski, "Erosion, dilation and related operators," *International Mathematics Symposium, 8th International Mathematica*

Symposium, June 2006.

[18] Francisco P. M. Oliveira, Faculdade de Engenharia, and João Manuel R. S. Tavares, "Matching Contours in Images through the use of Curvature, Distance to Centroid and Global Optimization with Order-Preserving Constraint," *CMES*, vol. 43(1), pp. 91-110, 2009.

[19] Akondi Vyas, M B Roopashree, and B Raghavendra Prasad, "Centroid Detection by Gaussian Pattern Matching in Adaptive Optics," *IJCA*, vol. 1(26), pp. 30-35, 2009.

[20] Tassos Markas, "Quad Tree Structures for Image Compression Applications," *Information Processing & Management*, vol. 28, no. 6, pp. 707-721, 1992.

[21] Algorithmic Graph theory David Joyner, Minh van Nguyen and Nathan Cohen 2011 ebook, http://code.google.com/p/graph-theory-algorithms-book.

Surabhi Narayan is currently Associate Professor in the department of Computer Science & Engineering at B.N.M Institute of Technology Bangalore. She has completed her B.E from SJCE, Mysore and MTech from University of Mysore. She is currently pursuing her research in the area of Document Image Processing in Visvesvaraya Technological University. Her research interests include Image Processing and Pattern Recognition.

Sahana D Gowda is currently Professor & Head, in the Department of Computer Science & Engineering at B.N.M Institute of Technology, Bangalore. She completed her PhD from University of Mysore in the year 2009. Her research interests include Image Processing, Pattern Recognition, and Big Data Analytics. She has published more than 25 papers in various International Conferences and Journals. She has many funded projects to her credit.