Rapid Determination of Co-occurrence Texture Features

David A. Clausi, Yongping Zhao
Department of Systems Design Engineering
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
Email : dclausi@uwaterloo.ca, Tel: (519) 888 – 4567 x2604

Abstract-Typically, the co-occurrence features for image processing are calculated by using a grey level co-occurrence matrix (GLCM). This method is computationally intensive since the matrix is usually sparse leading to many unnecessary calculations involving zero probabilities. An improvement on the GLCM method is to utilize a grey level co-occurrence linked list (GLCLL) to store only the non-zero co-occurring probabilities. The GLCLL suffers since, to achieve preferred computational speeds, the list should be sorted.

This paper presents a grey level co-occurrence hybrid structure (GLCHS) based on an integrated hash table and linked list approach. Texture features obtained using this technique are identical to those obtained using the GLCM and GLCLL. Based on a Brodatz test image, the GLCHS method is demonstrated to be a superior technique when compared across various window sizes and grey level quantizations. The GLCHS method required, on average, 33.4% of the computational time (σ = 3.08%) required by the GLCLL. Significant computational gains are made using the GLCHS method.

Index Terms-Texture features, hash table, linked list, co-occurrence probabilities, remote sensing imagery.

I. INTRODUCTION

Grey-level co-occurrence matrices (GLCMs), developed by Haralick et al. [1] are widely used in image texture feature extraction for spatial analysis of remotely sensed imagery [2], [3], [4]. A primary computational drawback of GLCMs is that they require unnecessarily high computational requirements when applying the statistics, which leads to an overwhelming amount of computation when attempting to segment full remote sensing images.

There are a number of approaches to reduce the computational requirements when calculating texture features when using GLCMs [5], [6], [7]. For example, the grey level co-occurrence linked list (GLCLL) method achieves a significant reduction computational requirements [7]. This is achieved since, unlike the GLCMs, the GLCLLs use a linked list to store only the non-zero co-occurring probabilities.

To efficiently access grey level pairs on the linked list to update their probabilities, the list is kept sorted. This sorting compromises the efficiency of the GLCLLs. In this paper, a grey level co-occurrence hybrid structure (GLCHS) based on an integrated hash table and linked list approach is presented. This algorithm shows a significant and consistent improvement in computational performance relative to GLCLLs. The texture features captured by each of these three methods are identical.

2	2	2	3	3
3	2	2	3	3
2	2	2	2	2
3	0	0	2	2
0	0	0	0	2

a

i \ j	0	1	2	3
0	0.2	-	-	-
1	0	0	-	-
2	0.1	0	0.4	-
3	0.05	0	0.15	0.1

b

Fig. 1. a. 5 x 5 image window with four grey levels values (0-3)
b. Corresponding GLCM given δ =1 pixel and θ = 0° and 180°. Only lower triangle matrix required.

II. EXISTING CO-OCCURRENCE IMPLEMENTATIONS

The GLCM technique employs the following steps. The probability of co-occurrence between two grey levels i and j given a relative orientation (θ) and distance (δ) can be computed for all possible co-occurring grey level pairs in an image window. The GLCM stores these probabilities and, as such, is dimensioned to the number of grey levels (G) available (Fig. 1). Then, selected statistics are applied to the GLCM by iterating through the entire matrix (ie. over all probabilities) to calculate the texture features.

Dissimilarity, contrast, uniformity, entropy, and correlation are five statistics widely used among the fourteen original statistics developed by Haralick et al. [1]. Several authors have discussed their meanings [2], [3]. For each unique (δ, θ) pair, a GLCM (or its equivalent) is required (Fig. 1).

In general, relative interpixel distances are short when applied to remotely sensed imagery and often only setting δ = 1 is required to generate preferred texture features [5]. The relative interpixel orientation is usually set to either {0°, 45°, 90°, 135°} or the average of all four orientations. Co-occurring pairs oriented at 0° are also oriented at 180° which generates a symmetrical GLCM. This concept extends to 45°, 90°, and 135° as well. As a result, only the lower triangular

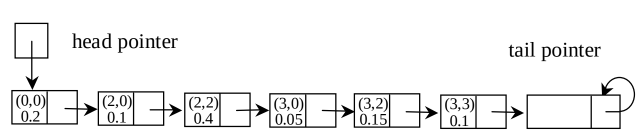

Fig. 2. GLCLL structure for determining image texture features. To reduce search times, the nodes are sorted according to grey level pairs (i,j).

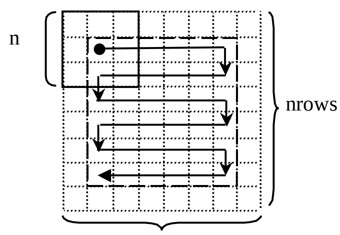

Fig. 3. Zig-zag window path to determine texture features for entire image.

portion of the GLCMs (ie. only co-occurring pairs where i >= j) needs to be retained (Fig. 1b).

The generation of co-occurrence probabilities is illustrated in Fig. 1. Here, given $G = 4$, $\theta = 0^{\circ}$, and $\delta = 1$ pixel, co-occurring probabilities are generated for a 5x5 sample window (Fig. 1a) and represented within a symmetrical GLCM (Fig. 1b). For example, the co-occurring pair (2,3) occurs three times in the window. Since there are a total of 20 possible co-occurring pairs, this generates a probability of 0.15 at row 3 and column 2 in the GLCM. With increasing *G*, the matrix becomes sparser with a corresponding increase in the computational requirements to apply the statistics.

Instead of using a matrix to store the co-occurrence probabilities, a linked list structure (grey level co-occurrence linked list or GLCLL) can be used [7]. In order to allow rapid searching for existing (i,j) pairs, the list must be kept sorted. Searching begins at the head of the list by looking for the first instance of the i^{th} grey level. If found, then the algorithm searches for the corresponding j^{th} grey level. If the (i,j) pair is found, then the probability stored inside that node is incremented. If the (i,j) node is not found at the expected location, then a node must be entered at that location that stores the appropriate probability for that grey level pair.

When capturing texture features on a pixel-by-pixel basis from an image, a sliding window is employed. Since most probabilities remain the same when the window shifts one column, the algorithm only has to include the new probabilities introduced by the new column and account for the probabilities that the window has just moved past. The window moves efficiently in this zig-zag pattern until the entire image is covered (Fig. 3).

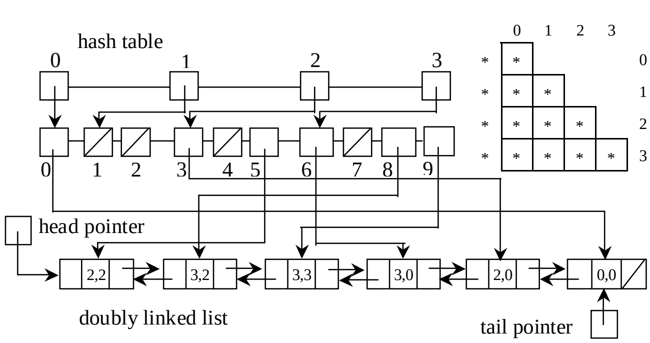

Fig. 4. GLCHS structure for determining image texture features. Nodes created in linked list and hash table are based on sample image in Fig. 1A.

Considerable time is spent maintaining sorted linked lists. The application of the statistics to calculate the texture features occurs rapidly by traversing each linked list from head to tail.

III. Hybrid Data Structure

To improve on the GLCLL, the reliance on maintaining sorted lists should be removed. A combined hash table and linked list structure is designed to meet this criterion. Based on this combination, the structure is referred to as the grey level co-occurrence hybrid structure (GLCHS). Using the GLCHS, a two-dimensional hash table **struct** is created to point to linked list nodes. A doubly linked list is used to allow easy insertion and deletion of nodes. The hash table allows for rapid access of any node in the linked list, if that node exists. The linked list allows for rapid application of the statistics by traversing the linked list from head to tail. The C **struct** definition for nodes in the linked list is:

```
Typedef struct ListNode
{
   int x1,x2;  //  co-occurring grey level pairs
   struct ListNode *prev;
   struct ListNode *next;
 } ListNode;
```

and the stuct definition for nodes in hash table is:

```
Typedef struct HashNode
 {
    float pr;    //co-occurrence probability
    struct ListNode *list_ptr;
 } HashNode;
```

In the ListNode **struct**, four members are defined. Each instance of the **struct** represents a node on the doubly linked list. The two integer members (**x1**,**x2**) store the grey level pairs. Two self-referential pointers are defined to access previous (***prev**) and next (***next**) ListNode nodes. Linked list nodes are defined to represent the first (**head**) and the last (**tail**) nodes.

In the hash table **struct**, one float member (**pr**) stores the grey level co-occurrence probability and the other stores the linked list pointer (***list_ptr**). The **list_ptr** points to the corresponding node on the linked list associated by the grey level pair.

Based on the two definitions, the creation of the hybrid data structure requires the following steps (Fig. 4). First, dynamic memory allocation is used to create the hash table. Second, a double pointer is set as a list of pointers to the rows in the hash table and all of the nodes are initialized (**pr** set to zero and pointer set to NULL). Finally, the head and tail are initialized to appropriate values to represent an empty doubly linked list.

The co-occurring pair is forced to have the relationship x1>=x2 so that only a lower triangular hash structure is required. For a given grey level pair, if the hash table has a zero entry, then that particular co-occurring pair does not

TABLE I
PERCENTAGE RATIOS BASED ON RESULTS FOR GLCHS VS. GLCLL

Comparison [%]	**Grey Levels** (G)	**Window Size** (n x n)					
		5 x 5	**10 x 10**	**15 x 15**	**20 x 20**	**25 x 25**	**30 x 30**
Percentage of GLCHS computational time compared to GLCLL computational time	**128**	37.5	35.4	34.9	32.1	30.3	28.2
	64	38.0	35.1	33.3	30.9	30.2	28.1
	32	37.5	34.8	32.7	32.5	29.0	28.9
	16	37.2	34.2	33.0	34.4	29.8	29.1
	8	37.7	36.3	35.3	35.6	34.7	34.3

have a representative node on the linked list. As a result, a new ListNode is created, its grey level values are set, and it is inserted at the end of the linked list. The **list_ptr** is then set to point to this ListNode to establish the relationship between the HashNode and its corresponding ListNode. If the hash table entry is not zero, then that HashNode already points to an existing ListNode on the linked list. Whether or not the ListNode was created, the probability associated with HashNode is incremented by the given probability. A similar method is used to decrement a probability for a certain grey level pair. In this situation, if the probability reaches zero, the ListNode is removed from the linked list and its associated HashNode's **list_ptr** is set to NULL. Fig. 4 illustrates the structural arrangement for the GLCHS. As a result, the linked list does not have to be kept sorted, in contrast to the GLCLL. This design is expected to significantly and consistently reduce the completion times when determining co-occurrence probability texture features.

IV. RESULTS ANALYSIS

All algorithms are implemented using the C language in a Unix environment based on the same fundamental code ie. the only distinctions between the routines are the algorithms themselves. The tests are performed on the Sun Sparc Ultra 1 200E (200 MHz, 128 Mbytes RAM, 322 SPECint, 462 SPECfp) computer Workstation with a multi-class 128 x 128 Brodatz [8] test image. Table 1 contains the percentage ratios of numerical times in capturing texture features for a single window between the two algorithms with the five statistics referred in the beginning of Section II [1], [2]. The parameters are: $\delta = 1$; $\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$; six window sizes (5, 10, 15, 20, 25, and 30 pixels); and five quantization grey levels (128, 64, 32, 16, and 8).

The results show that the GLCHS is always significantly faster than the GLCLL. The greater the number of grey levels, the greater the improvement of the GLCHS over the GLCLL method. For example, given a window size of 30x30 pixels, the ratio between GLCHS and GLCLL at 128 grey levels is 28.2% while the ratio at 8 grey levels is about 34.3%.

The window size impacts the length of the linked lists which in turn affects the computational speeds. For increasing window size, the number of co-occurring probabilities will generally be increased, so the time spent on determining statistics (both GLCLL and GLCHS) and determining the probabilities (GLCLL only) will be increased either. For example, when $G = 128$ grey levels in Table I, the ratio between GLCHS and GLCLL for 5x5 windows is 37.5% and, with increased window size, gradually decreases to 28.2% for window size 30x30. The advantage of GLCHS over GLCLL improves with larger window sizes.

V. CONCLUSIONS

Image texture segmentation is often performed on remotely sensed imagery, and co-occurrence probabilities are commonly used for feature extraction. The advantage of the GLCHS methodology for determining co-occurrence texture features relative to the GLCLL method is clearly demonstrated in the results (Table 1). With larger images (typical of remote sensing imagery), the computational impact of using the GLCHS algorithm is extremely important. Granted, the computational savings will be a function of the textural characteristics, window size, number of statistics, and quantization level. However, on average across all the test cases, GLCHS required 33.4% ($\sigma = 3.08$) of the computational time compared to GLCLL, which strongly supports its use.

ACKNOWLEDGEMENTS

Funding for this project was provided by Geomatics for Informed Decisions (GEOIDE), a Network of Centres of Excellence (NCE) supported by the Canadian funding agency Natural Sciences and Engineering Research Council (NSERC) (http://www.geoide.ulaval.ca).

REFERENCES

[1] R.M. Haralick, K. Shanmugam, and I. Dinstein, "Texture features for image classification," *IEEE Trans. on Syst. Man Cybern*., vol. 3, pp. 610-621, Nov. 1973.

[2] M.E. Shokr, "Evaluation of second-order texture parameters for sea ice classification from radar images," *J. Geophys. Res.*, vol. 96, pp. 10625–10640, no. 6 1991.

[3] A. Baraldi and F. Parmiggiani, "An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters," *IEEE Trans. Geosci. Remote Sensing*, vol. 33, pp. 293-304, Mar. 1995.

[4] L.-K. Soh and T. Tsatsoulis, "Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices," *IEEE Trans. on Geosci. Remote Sensing*, vol. 37, pp. 780-795, Mar. 1999.

[5] D.G. Barber and E.F. LeDrew, "SAR sea ice discrimination using texture statistics: a multivariate approach," *Photogramm. Eng. Remote Sensing*, vol. 57, pp. 385-395, Apr. 1991.

[6] I.S. Duff, A.M. Erisman, and J.K. Reid, *Direct Methods for Sparse Matrices*. Oxford, London: Clarendon Press, 1986.

[7] D.A. Clausi and M.E. Jernigan, "A fast method to determine co-occurrence texture features," *IEEE Trans. Geosci. Remote Sensing*, vol. 36, pp. 298-300, Feb. 1998.

[8] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, New York: Dover, 1966