MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

USING AN OCTREE-BASED RAG IN HYPER-IRREGULAR PYRAMID SEGMENTATION OF TEXTURE VOLUME

Horace H. S. Ip and Stephen W. C. Lam
Image Computing Group, Department of Computer Science, City Polytechnic of Hong Kong

Tat Chee Ave. Kowloon, Hong Kong
Email :cship@cphkvx.cphk.hk

ABSTRACT

We extend the irregular pyramid segmentation scheme to three dimensional (3D) digital space and we call it hyper-irregular pyramid segmentation. Based on this segmentation scheme, a 3D texture image is split into partitions recursively based on octree structure. The texture features are calculated based on 3D gray level spatial dependency measurement. The octree is subsequently converted into a 3D region adjacency graph (**RAG**). Each vertex of the graph consists of a texture feature vector of the corresponding partition and each edge represents the neighborhood relationship between two partitions. The hyper-irregular pyramid process is then applied to the graph and finally a segmentation of the three dimensional image is obtained.

INTRODUCTION

Three dimensional segmentation techniques are generally based on extending 2D techniques into 3D digital space and can be broadly classified into two categories. They are volume based and edge based approaches. The edge based approach detects 3D discontinuity to delineate a three dimensional volume while volume based approach extracts volume as a whole. Texture feature extraction has been attempted in numerous ways. Surveys can be found in [10]. In this research, we extract three dimensional texture features based on some easily computable gray level spatial dependencies [5]. Traditionally, pyramid architecture construction is based on rigid local subsampling scheme [2][3][4]. Recently, adaptive or irregular pyramid has been proposed for image segmentation [6][7]. Under this new framework, the clustering of pixels are treated as a graph contraction problem. In this paper, we extend the irregular pyramid to 3D space for segmenting 3D texture image.

REGION ADJACENCY GRAPH CONSTRUCTION

As stated in [11], a region adjacency graph (**RAG**) is a description of the homogenous regions of an image and their neighboring relationships. In this section, we present the details of forming **RAG** for 3D texture images.

Octree Splitting

During this stage, the image volume is split up into eight partitions recursively based on an octree structure. The root of the octree corresponds to the whole image volume and each node of the octree represents a partition of the image volume. We define the initial image volume to be composed of $2^n \times 2^n \times 2^m$ voxels where $m < n$. In each level except level 0, the volume of the partition of the corresponding node is 1/8 of that of its father. That is, in level L, the partitions are of dimension $2^{n-L} \times 2^{n-L} \times 2^{m-L}$.

Let X^P={ $X_{\mathbf{NWF}}$, $X_{\mathbf{NEF}}$, $X_{\mathbf{SWF}}$, $X_{\mathbf{SEF}}$, $X_{\mathbf{NWB}}$, $X_{\mathbf{NEB}}$, $X_{\mathbf{SWB}}$ and $X_{\mathbf{SEB}}$ } denotes the set of partitions of a volume X and X^V be the texture feature vector of a partition X. The following pseudo code describes the recursive splitting process.

Initially, x represents the whole image volume.

```
Splitting_test( x )
{
        if ( ∃x_i ∈x^P | δ(x_i^V, x^V) > H )
        {
           x is split up into eight partitions;
           ∀x_i ∈x^P, do Splitting_test( x_i );
        }
        else return;
}
```

where δ denotes the Euclidean distance function and **H** is a threshold value. The calculation of the texture feature vector is shown in the next section.

Through this process, the three dimensional image is split into eight partitions recursively until no more heterogeneous texture partition is detected. The calculation of texture features is described in the following section.

Texture Feature Extraction

Co-occurrence matrices which describe the gray level spatial dependency in two dimensional image has been presented in [5]. In this research, we extend it to represent 3D gray level spatial dependency. Suppose the image partition to be analyzed is of dimension $2^n \times 2^n \times 2^m$ and the gray level appearing in each voxel is quantized to N_g levels. Let $L_x = \{0, 1, \ldots, 2^n-1\}$ and $L_y = \{0, 1, \ldots, 2^n-1\}$ be the horizontal spatial domains, $L_z = \{0, 1, \ldots, 2^m-1\}$ be the vertical spatial domain, and $G = \{0, 1, \ldots, N_g-1\}$ be the set of N_g quantized gray levels. The three dimensional image **I** can be represented as a function which assigns some gray level in G to each voxel or coordinates in $L_x \times L_y \times L_z$; $\mathbf{I}: L_x \times L_y \times L_z \to G$.

Similar to [5], in order to extract such spatial relationship, we construct gray level co-occurrence matrices which record the relative frequencies $P^{\omega}(i,j)$ with which two neighboring voxels in the image volume, one with gray level i and the other with gray level j in direction ω where $\omega=0,1,..,8$. There are nine co-occurrence matrices and each matrix measures the spatial dependency of one direction. The definition of the co-occurrence matrices in nine directions are shown in the following:

Given $((k,l,p),(m,n,q)) \in (Lx \times Ly \times Lz) \times (Lx \times Ly \times Lz)$,

$\mathbf{I}(k,l,p) = i$ and $\mathbf{I}(m,n,q) = j$.

$$P^0(i,j) = \#\{((k,l,p),(m,n,q)) | l-n=0, p-q=0, |k-m| = d\}$$

$$P^1(i,j) = \#\{((k,l,p),(m,n,q)) | k-m=0, p-q=0, |l-n| = d\}$$

$$P^2(i,j) = \#\left\{\begin{matrix}((k,l,p),(m,n,q)) | p-q=0, (k-m=d, l-n=-d) \\ \text{or } (k-m=-d, l-n=d)\end{matrix}\right\}$$

$$P^3(i,j) = \#\left\{\begin{matrix}((k,l,p),(m,n,q)) | p-q=0, (k-m=d, l-n=d) \\ \text{or } (k-m=-d, l-n=-d)\end{matrix}\right\}$$

$$P^4(i,j) = \#\{((k,l,p),(m,n,q)) | k-m=0, l-n=0, |p-q| = d\}$$

$$P^5(i,j) = \#\left\{\begin{matrix}((k,l,p),(m,n,q)) | (k-m=d, l-n=d, p-q=d) \\ \text{or } (k-m=-d, l-n=-d, p-q=-d)\end{matrix}\right\}$$

$$P^6(i,j) = \#\left\{\begin{matrix}((k,l,p),(m,n,q)) | (k-m=d, l-n=d, p-q=-d) \\ \text{or } (k-m=-d, l-n=-d, p-q=d)\end{matrix}\right\}$$

$$P^7(i,j) = \#\left\{\begin{matrix}((k,l,p),(m,n,q)) | (k-m=d, l-n=-d, p-q=-d) \\ \text{or } (k-m=-d, l-n=d, p-q=d)\end{matrix}\right\}$$

$$P^8(i,j) = \#\left\{\begin{matrix}((k,l,p),(m,n,q)) | (k-m=d, l-n=-d, p-q=d) \\ \text{or } (k-m=-d, l-n=d, p-q=-d)\end{matrix}\right\}$$

where d = 1 and # denotes the number of element in the set.

The above nine co-occurrence matrices are constructed for each partition. For each co-occurrence matrix, we compute three measures and they are: f_1^{ω} (second angular moment)(**ASM**), f_2^{ω}(contrast) and f_3^{ω}(correlation) which are calculated by the following equations:

$$f_1^{\omega} = \sum_{i=0}^{Ng-1}\sum_{j=0}^{Ng-1}\left(\frac{P^{\omega}(i,j)}{R}\right)^2$$

$$f_2^{\omega} = \sum_{n=0}^{N_g-1} n^2 \left\{\sum_{|i-j|=n}\left(\frac{P^{\omega}(i,j)}{R}\right)\right\}$$

$$f_3^{\omega} = \frac{\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}\left[ij\,P^{\omega}(i,j)/R\right] - \mu_x\mu_y}{\sigma_x\sigma_y}$$

where $\mu_x, \mu_y, \sigma_x, \sigma_y$ are the means and standard deviation of p_x and p_y and

$$p_y(j) = \sum_{i=0}^{N_g-1} p^{\omega}(i,j)$$

and $\quad p^{\omega}(i,j) = P^{\omega}(i,j)/R$

where R is the number of voxel pairs used in calculating a particular gray level co-occurrence matrix. The feature vector describing a partition is calculated by taking the average of the spatial dependency measurement of all the sampling directions. That is, a feature vector $[f_1, f_2, f_3]$, with each feature f_i defined by

$$f_i = \frac{\sum_{\omega=0}^{8} f_i^{\omega}}{9} \qquad \text{where } i=1,2,3$$

Each f_i represents the texture characteristic of a partition

Tree to graph conversion

From the octree, we already have leaf nodes indicating the locations of various partitions with homogenous texture and their corresponding feature vectors. So, we have enough information to construct the node set of a **RAG**. To compute the edge set, we need to know the neighboring relationship among these partitions. In [11], the neighboring relationship among the partitions are found by first defining the following functions:

a). **ADJ**(O,s) is *true* if octant O_n, denoting the octant node n fills in his father, is adjacent to side s, the direction in which we search for neighbors,

b). **REFLECT**(O,s) indicates which octant on the same level borders to side s of octant O.

c). **DIRECT**(s) supplies the octants that are not adjacent to side s.

Using these functions it is easy to find every adjacent neighbor in any of the six directions. For detail description of applying the three functions for neighbor finding, please consult [11]. After this conversion process, the neighbor relationship among the leaf nodes are known. A **RAG** can then be constructed with each vertex of it describing the texture characteristics of a partition and each edge represents the neighbor relationship between two adjacent partitions.

HYPER-IRREGULAR PYRAMID CONSTRUCTION

Sequential region growing algorithms have been proposed for grouping regions based on **RAG** [1][8][9]. Here we present a hyper-irregular pyramid construction process. This process groups regions together parallelly using **RAG.**

Following the convention stated in [11], each layer of the pyramid is a graph $G[l]=(V[l], E[l])$ where $V[l]$ is the set of vertices and $E[l]$ is the set of edges at level l. $G[0]$ is the **RAG** representing the texture volume. The vertex set at layer l (i.e. $V[l]$) is the set $\{v_i[l]\ i=0..n-1\}$ where n is the number of vertex at layer l. The basic component of each vertex, as we have mentioned before, is a feature vector $\boldsymbol{f}$. We now associate each vertex a variable $\boldsymbol{x}$. $\boldsymbol{x}$ is a random variable uniformly distributed between [0,1]. In the following, we adopt the convention of "a.b", where *a* represents a vertex and *b* an attribute of the vertex. For instance, $v_i[l].x$ represents the variable x associated with vertex i at level l. Furthermore, $\{n_k(v_i[l])\ \forall k = 0..u-1\}$ is the set of the neighboring vertices of $v_i[l]$ where u is the number of neighboring vertex of $v_i[l]$ and $n_0(v_i[l]) = v_i[l]$. The neighborhood relation is defined between two vertices if they share a common edge.

The decimation process from layer l to layer l+1 is presented as follows:

Initially, $l = 0$,

1) Generate random number x for all vertices.
2) A vertex is selected as a survivor if
a) $v_i[l].\boldsymbol{x} = \max(\{\mathbf{n}_k(v_i[l]).x\ \ \forall k=0..u-1\})$ or
b) $\min(\delta(\ \mathbf{n}_k(v_i[l]).\boldsymbol{f},\ \ v_i[l].\boldsymbol{f}\ )\ \ \forall k=1..u-1\ ) > \mathbf{T}$
3) if a non-survivor cannot find a neighboring survivor, goto 1.
4) Duplicate all the survivors and put them to layer l+1.

In general, a vertex is selected as a survivor if its random variable x is local maximum among its neighbors or its feature vector differs a lot from its neighbors. After the decimation process, we now consider the formation of the edge set. An edge set at level l+1 is related to the neighborhood relationship of the tessellation volumes at level l. As stated in [5], the edge formation scheme is a survivor expanding process which includes the following two steps:

Assume vertex i has been selected as a survivor:

1. All the vertices of the graph sharing an edge with vertex i are analyzed. If a nonsurvivor is not claimed by any of the other survivors it is incorporated into the vertex i. If a set of survivors S competes for a non-survivor, the non-survivor allocates itself to a survivor n given that difference of their shape feature vectors is minimum for all the survivors in S.

2. The vertex i tries to expand from the vertices already incorporated into it. The expansion is stopped whenever a survivor or an already-allocated nonsurvivor is met.

For convenience, we name the set of vertices occupied by a survivor including the survivor itself as *territory*. The edges of the pyramid at layer l+1 can now be defined based on the adjacency of the territories at layer l. If two territories are neighbors at layer l, the two vertices at layer $l + 1$ corresponding to the territories' center vertices (i. e. the two expanding survivors at layer l) are linked together by an edge. After this edge formation process, the son to father linkages can also be defined. As survivors at layer l are duplicated to layer l+1, they are simply linked to their corresponding vertices at layer l+1. For those non-survivors, they are linked to the vertices corresponding to the survivors which they are incorporated to during the survivor expanding process carried out at layer l.

Finally, the shape feature vectors of the vertices at layer l+1 is calculated by taking the average of the shape feature vectors of its sons at layer l. That is,

$$v_i[l+1].\boldsymbol{f} := \mathrm{ave}(\ M\ ) \text{ where } M = \{\ v_j[l].\boldsymbol{f}\ \mid v_j[l] \text{ is a son of } v_i[l+1]\}$$

The pyramid construction process stops at layer L when $G[L] = G[L-1]$.

EXPERIMENTAL RESULTS

One texture image sequence is used for testing our hyper-irregular pyramid segmentation scheme. The texture image sequence used for the experiment is composed of 64 128×128 images with 16 gray levels. The image sequence consists of two distinct texture volumes. For this example, an octree is formed down to level 2. Figure 1 shows the image data and the segmentation results.

CONCLUSION

Our approach provides a general framework for three dimensional image segmentation. The feature extraction technique is not restricted to gray level dependency measurement. Other texture feature extraction techniques can also be used with this scheme. Although the segmentation scheme presented in this paper is developed for segmenting 3D disordered texture, the hyper-irregular pyramid can be used to segment structural texture image volume. This can be done by changing each vertex into feature vector representing the shape of a texture token. The same hyper-irregular pyramid construction process can be applied to segment the image volume.

REFERENCE

[1] Browning, J. S. and S. L. Tanimoto, Segmentation of Pictures into Regions with a Tile-by-tile Method, Pattern Recognition 15, 1-10 (1982).

[2] P. J. Burt, T.H. Hong. and A. Rosenfeld, Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-11, No.12, 802-809, (1981).

[3] J. Cibulskis and C.R.Dyer, Node Linking strategies in pyramids for segmentation. *Multiresolution Image Processing and Analysis*, edited by A. Rosenfeld. Berlin: Springer-Verlag, 109-120, (1984).

[4] T. H. Hong and A. Rosenfeld, Compact region extraction using weighted pixel linking in a pyramid *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, 222-229 (1984).

[5] R. M. Haralick, K. Shanmugam and Its'hak Dinstein, Textural Features for Image Classification, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-3, No. 6, November, 610-621, (1973).

[6] J. M. Jolion and A.Montanvert, The Adaptive Pyramid: A Framework for 2D Image Analysis. *CVGIP: Image Understanding*, Vol. 55, No. 3, May, 339-348 (1992).

[7] A. Montanvert, Peter Meer and Azriel Rosenfeld, Hierarchical Image Analysis Using Irregular Tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 4, 307-316, (1991).

[8] Patrick C. Chen and Theodosios Pavlidis, Segmentation by Texture Using a Co-Occurence Matrix and a Split-and-Merge Algorithm, Computer Graphics and Image Processing Vol. 10, No. 2, June 1979.

[9] Pavlidis T., Structural Pattern Recognition, Springer, Berlin (1977).

[10] T. R. Reed, J. M. Hans Du Buf, A Review of Recent Texture Segmentation and Feature Extraction Techniques, *CVGIP: Image Understanding*, Vol. 57, No. 3, May, 359-372, (1993).

[11] K. C. Strasters, J. J. Gerbrands, Three-Dimensional Image Segmentation Using a Split, Merge and Group Approach, *Pattern Recognition Letters 12*, 137-320, (1991).

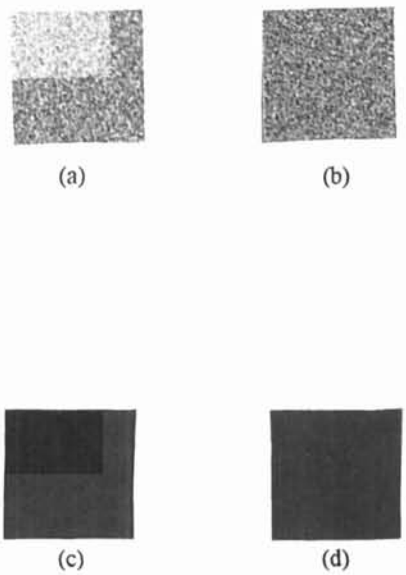

Figure 1. Segmentation for three dimensional random texture image. (a) One slice of the upper half image volume, (b) One slice of the lower half image volume, (c)(d) The corresponding segmentation result of (a) and (b).