A large number of volumetric data are available and are used for various scientific fields today. This paper describes a method to analyze volumetric data by a technique based on fractal dimension analysis. In the analysis, Hurst operators are used for estimating fractal dimension of volumetric data. A comparison between a 2D Hurst operator and a newly extended 3D Hurst operator based method is performed on volumetric data for our preliminary experiments. The experiment shows that both the 2D and 3D Hurst operators successfully segment patterns of volumetric data. It appears that the 3D method is slightly superior to the 2D method for segmenting certain type patterns.

1. Introduction

A large number of volumetric data are used for various scientific fields reflecting recent advancements in computer graphics processors. Pattern feature extraction techniques for volumetric data serve an important function in various pattern recognition and computer vision applications. Although intensive research has been conducted for extracting pattern features from 2D image data, extraction techniques for 3D volumetric data have not been investigated sufficiently. Recently, pattern feature extraction techniques for volumetric data have been proposed which include extension of a run-length encoding [1], extension of co-occurrence matrices [2], extension of higher order local autocorrelations [3], and extension of LAWS masks [4]. These techniques originated from traditional 2D texture image analysis techniques, but they are different in that they can handle three dimensional volumetric data. Since these recently introduced techniques directly extract pattern features from 3D volumetric data without slicing the data into multiple 2D images, the techniques can avoid loss of pattern features caused by slicing processes. In this research, we apply a three dimensional extension technique to the Hurst operators [5] for
fractal dimension analysis. The idea is quite similar to the four techniques mentioned above, and the extension enables precise handling of 3D volumetric data.

2. **Shape Feature Extraction Based on Hurst Operators**

In this section, (1) Fractal Dimensions and (2) Hurst Operators are discussed.

2.1. **Fractal Dimension**

Fractal based techniques have been used to discriminate 2D texture patterns for various applications in computer visions and recognitions. In the techniques, fractal dimension is estimated for measuring irregularity of 2D images, and it can characterize texture patterns for segmentations. There are various techniques for determining the fractal dimension of 2D images, which include a box counting technique [5] and a Hurst coefficient computation technique [6]. The Hurst coefficient computation technique is one of the important techniques that has been used for various segmentation applications. Paper [6] shows the Hurst measurement of 2D images, and in the analysis technique of the authors, two dimensional circular filters were used for computing Hurst coefficients for estimating fractal dimensions of 2D images.

2.2. **3D Hurst Operators**

In our research, since we are dealing with 3D volumetric data, the two dimensional Hurst operators were extended to the three dimensional Hurst operators. In other words, the 2D circular filters were extended to the 3D spherical filters. This extension makes it possible to extract pattern features directly from 3D volumetric data without slicing into multiple 2D images. Since the spherical filter should be able to change its diameter size for adapting the volumetric data analyzed, a simulation software program has been implemented which generates the various sizes of spherical filters. Figure 1 shows a spherical filter with a 5 voxel wide neighborhood, and Figure 2 shows the corresponding distance of voxels from the center of the neighboring voxels. In Figure 1, the spherical filter is sliced into 5 circular filters for convenience. Alphabetical labels identify classes of the same distance from the central voxel. By using the spherical filter, a list of the maximum grey level differences for each distance class of voxels is computed. Once the list is computed, the log of both the distance and the grey level difference is taken for computing a least square fit line. The slope of the line indicates a Hurst coefficient of the
volumetric data. This process is repeated for computing the Hurst coefficients for each voxel location in the volumetric data. New volumetric data can be created by storing the coefficients for those voxel locations by using the obtained Hurst coefficients. This newly created volumetric data can be referred to as the Hurst transform volume (i.e., the Hurst transform image for the 2D image case). Often the Hurst transformed data can be useful for visualizing patterns that are difficult to detect in the original data. Since the data can detect the irregularity of patterns, it is useful for applications which require segmentation of data. Details of the Hurst coefficients computation processes for 2D images can be found in papers [7][8], and the computation processes described above are quite similar to those of the 2D image case except that the computation of the maximum grey level differences are examined not only in the x and y directions but also in the z direction. The diameter of the filter can vary depending on the applications. In our experiments, diameters of 3, 5, 7, 9, 11 and 13 (voxels) were examined. Basically, a filter with a large diameter size can estimate more accurate Hurst coefficient values. However, a filter with a large diameter size requires a considerable amount of computation time, and it introduces unprocessed regions at the outer boundary of volumetric data.

![Fig. 1: Spherical filter with a 5 voxel wide neighborhood and the corresponding distance of voxels from the center of the neighboring voxels.](image)

<table>
<thead>
<tr>
<th>Label</th>
<th>Distance</th>
<th>Num.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.000000</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>1.000000</td>
<td>6</td>
</tr>
<tr>
<td>c</td>
<td>1.414214</td>
<td>12</td>
</tr>
<tr>
<td>d</td>
<td>1.732051</td>
<td>8</td>
</tr>
<tr>
<td>e</td>
<td>2.000000</td>
<td>6</td>
</tr>
<tr>
<td>f</td>
<td>2.236068</td>
<td>24</td>
</tr>
<tr>
<td>g</td>
<td>2.449490</td>
<td>24</td>
</tr>
</tbody>
</table>

![Fig. 2: Distance of voxels from the center of the neighboring voxels.](image)
3. Experiments and Results

The segmentation method using the 2D Hurst operators was compared to the method using the 3D Hurst operators. In the experiment, artificially synthesized 3D solid textures based on Perlin noise functions [9] were used as samples. Various operator sizes (3, 5, 7, 9, 11 and 13) were examined for computing the Hurst coefficients for each set of experimental data. Three types of solid textures including (1) cloud, (2) marble and (3) turbulence were used as shown in Figure 3. Each solid texture has sizes of 64x64x64. These solid textures can be sliced into 64 two dimensional images, and the 32nd slice was used for computing 2D Hurst coefficients. For the 3D Hurst coefficients computation, the 32nd slice and its neighboring slices were used. For example, the 3D Hurst operator with a size of 5 uses slices of the 30th, 31st, 32nd, 33rd, and 34th. When the sizes of the operator increase, the number of neighboring 2D slices increases.

Figures 4, 5, and 6 show three types of solid textures analyzed by 2D and 3D Hurst operators (Hurst transform image and Hurst transform volume). In the figures, each column represents the Hurst operator sizes, and each row represents (a) the 32nd 2D image, (b) the 2D Hurst transform image, (c) the 32nd 2D image sliced from the 3D Hurst transform volume, (d) the image which shows the difference between the 2D and 3D Hurst transform, (e) the equalized 2D Hurst transform image, and (f) the 2D image sliced from the equalized 3D Hurst transform volume.

As shown in the figures, both (b) the 2D Hurst transform image and (c) the 32nd 2D image sliced from the 3D Hurst transform volume showed quite similar results. However, in the case of (b) the 2D Hurst transform image, some noises (white or black spots) were detected for certain volume data. This happens especially when the analyzed volume data do not have a wide range of grey-level intensity values such as the example in "cloud" which is shown in Figure 4. In the case of 2D Hurst operators, there are not enough sample intensity data values for estimating Hurst coefficients. However, in the case of 3D Hurst operators, since multiple numbers of slices were used, more chances are available to use sample intensity data values for estimating Hurst coefficients, which avoids introduction of noises.

Both (e) the equalized 2D Hurst transform image and (f) the 2D image which sliced from the equalized 3D Hurst transform volume can be useful for visualizing particular patterns. In the cases of 'cloud' and 'turbulence' invisible patterns were revealed after applying 2D and 3D Hurst operators. Another interesting point is that particular patterns are extracted based on resolutions of the Hurst operators. In the case of 'marble', when the Hurst operator increases
in size, coarser marble patterns are observed while fine marble patterns are ignored.

Fig. 3: Three types of solid textures with sizes of 64x64x64 (cloud, marble, and turbulence).

Fig. 4 (left): Solid textures analyzed by 2D and 3D Hurst operators (Cloud).
Fig. 5 (right): Solid textures analyzed by 2D and 3D Hurst operators (Marble).

Table 1 shows the time required for computing the 2D and 2D Hurst coefficients. Also, the times required generating the 2D Hurst transform images and 3D Hurst transform volumes are shown in the table. Solid textures with sizes of 64x64x64 were used for evaluations, and the Hurst operators with sizes of 3, 5, 7, 9, 11 and 13 (voxels) were examined for each 2D and 3D Hurst operator. Ten solid textures were used for determining the average time. Although the time required for 3D Hurst coefficients and for generating the 3D Hurst transform volume increases rapidly for larger operator sizes, use of the large operator size makes it possible to obtain more accurate coefficients values.
Thus, there is a trade off between the computation time and the accuracy of the results.

![Image](image_url)

**Fig. 6 (left):** Solid textures analyzed by 2D and 3D Hurst operators.

**Table 1 (right):** Time (Sec.) required for computing the Hurst coefficients and for generating the Hurst transform volumes. (Pentium IV 3.4Ghz)

<table>
<thead>
<tr>
<th>Hurst Coefficients</th>
<th>Hurst Trans. Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>3D</td>
</tr>
<tr>
<td>3</td>
<td>0.001</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
</tr>
<tr>
<td>7</td>
<td>0.001</td>
</tr>
<tr>
<td>9</td>
<td>0.001</td>
</tr>
<tr>
<td>11</td>
<td>0.002</td>
</tr>
<tr>
<td>13</td>
<td>0.002</td>
</tr>
</tbody>
</table>

**4. Conclusion and Future Work**

In our research, 2D Hurst operators (circular filters) were extended to 3D Hurst operators (spherical filters) allowing the filters to handle 3D volumetric data. In the experiments, artificially synthesized 3D solid textures were examined by using both 2D and 3D Hurst operators. Our preliminary experiments showed that both 2D and 3D Hurst operators produce similar output (Hurst transform images and Hurst transform volumes). Also, we have found that 3D Hurst operators were able to analyze certain 3D volumetric data more accurately compared to the 2D Hurst operators, especially if the volumetric data do not contain a wide range of grey-level intensity values.

In our experiments, three types of artificially synthesized solid textures were examined. Additional solid textures which contain various patterns including three dimensional medical images will be examined in future works. Our current experimental system processes 3D volumetric data with a single processor. The experimental system will be modified for running software
programs concurrently with multiple numbers of processors so that the system can handle relatively large volumetric data more efficiently.

Acknowledgments

This research was partially supported by grants from the Telecommunications Advancement Foundation (TAF-2007) and the KAKENHI (19700108).

References


[06/2007a]