

Data Oriented Model of Image: as a Framework for Image Processing

A. Habibizad Navin, A. Sadighi, M. Naghian Fesharaki, M. Mirnia, M. Teshnelab, and R. Keshmiri

Abstract—This paper presents a new data oriented model of image. Then a representation of it, ADBT, is introduced. The ability of ADBT is clustering, segmentation, measuring similarity of images etc, with desired precision and corresponding speed.

Keywords—Data oriented modeling, Image, Clustering, Segmentation, Classification, ADBT and Image Processing.

I. INTRODUCTION

THE main idea of this paper is to present a data oriented model of image, and the abilities of it for clustering, segmentation, measuring similarity of images etc, with desired precision and corresponding speed are discussed.

Data oriented modeling is an approach which models concepts by using the data structure. We have introduced data oriented modeling of fuzzy controller for controlling the Anti-lock braking system [10]. Furthermore, we have modeled uniform random variables based on data oriented modeling [9]. In this paper, we represent data oriented model of images.

The Conventional methods use the color of the pixels to model the image and try to reduce the size of image (Compression).

Recently much research has been conducted on using data structures for performing image-processing tasks [1]-[7].

This paper characterizes a new model, which saves the image into a data structure too. This data structure is composed of several layers (called Levels). Each level is an image with specified resolution. Image processing tasks such as clustering, segmentation, measuring similarity, etc could be done with desired precision and corresponding speed by using related level of new model.

We have organized the paper as following: In section two, the conventional image formats such as GIF, BMP and JPEG are reviewed in brief. In the third section, data oriented model is introduced and using it to model the image. In section four

ability and aptitude of data oriented modeling to process the images such as measuring similarity, clustering, segmentation ... are discussed.

II. PREVIOUS IMAGE MODELS

Several formats have been used for storing images by now and the purpose of most of them is to reduce image sizes. Three most common conventional formats are reviewed:

A. BMP

BMP or .DIB (device-independent bitmap) is a bitmapped graphics format used internally by the Microsoft Windows and OS/2 graphics subsystem (GDI), and used commonly as a simple graphics file format on those platforms [11].

This format is in two parts, Part one, is the data, which indicate whole image features including width, height, color palette etc. The second part consists a block of bytes that describes the image, pixel by pixel. Pixels are stored starting in the bottom left corner going from left to right and then row by row from the bottom to the top. Each pixel is described using one or more bytes [11].

B. GIF

GIF (Graphics Interchange Format) is an 8-bit-per-pixel bitmap image format that was introduced by CompuServe in 1987 and has since come into widespread usage on the World Wide Web due to its wide support and portability. The format uses a palette of up to 256 distinct colors from the 24-bit RGB color space. GIF images are compressed using the LZW lossless data compression technique to reduce the file size without degrading the visual quality [12].

C. JPEG

In computing, JPEG (pronounced JAY-peg) is a commonly used standard method of compression for photographic images. The name JPEG stands for Joint Photographic Experts Group, the name of the committee who created the standard. The group was organized in 1986, issuing a standard in 1992, which was approved in 1994 as ISO 10918-1. The compression method is usually lossy compression, meaning that some visual quality is lost in the process, although there are variations on the standard baseline JPEG, which are lossless. There is also an interlaced "progressive" format, in which data is compressed in multiple passes of progressively higher detail. This is ideal for large images that will be displayed whilst downloading over a slow connection, allowing a reasonable preview before all the data has been retrieved [13].

A.Habibizad Navin is with the computer Engineering Department of Islamic Azad University, Science and Research Branch, Tehran, Iran. He is also collaborating with the IAUT Computer Research Laboratories, CRL (phone: +98-914 412 5973; fax: +98-411-3320725; e-mail: ManHabibi@yahoo.com).

A. Sadighi and R. keshmiri are with the Computer Engineering Department of Islamic Azad University of Tabriz, Iran. He is a member of ACM (e-mail: Sadighi.Aidin@gmail.com).

M. Naghian Fesharaki is with the Computer Department of Islamic Azad University, Science and Research Branch, Tehran, Iran.

M. Mirnia is with the Computer Engineering Department of Islamic Azad University, Tabriz Branch.

M. Teshnelab is with the Computer Department of Islamic Azad University, Science and Research Branch, Tehran, Iran.

Each of Mentioned formats has dedicated attributes, for example, BMP is the same as image, which represents it pixel by pixel but in GIF the LZW computing method with less number of bits per pixel are used to reduce the image size. In the same manner, JPEG uses a lossy compression method for reducing image size, which degrades the visual quality. In this paper a new model of image is introduced which image processing related algorithms can be executed on it with desired precision and corresponding speed. Data oriented modeling of image is explained in next section.

III. DATA ORIENTED MODELING

Data oriented modeling is an approach which models concepts with the aid of data structure, in the other words in this approach the data structures which can be simply and rapidly processed by computer, are used to model the concepts. A data oriented model named fuzzy-based-problem-solution data-structure, PSDS, has been introduced for controlling anti-lock-braking-system (ABS) [10]. In addition, two data oriented model, named uniform-digital-probability-diagraph, UDPD, and uniform-digital-n-complete-probability-tree, UDCPT, have been presented for modeling uniform random variables [9].

In this paper, similar to above models, a data oriented model of image is introduced. Recently much research has been conducted on using data structures for performing image-processing tasks. In [2] a binary-tree is used and a new algorithmic approach to segmentation-based image coding, named Adaptive-Tree-Structured-Segmentation, ATSS in short, is introduced. In [3] the peano scanning of an image is introduced as an alternative to the standard rasterization to produce a one-dimensional vector of image. In [1] a specific region-based-binary-tree representation of image incorporating with adaptive processing of the tree is proposed. Also several researches have been conducted on image compression using quad-tree representation of image [4]-[7]. Nevertheless, most of above researches use the data-structure-representation-of-image as a mean for doing a specific task such as segmentation or compression. I.e. those are not a general framework for image processing but a mean for performing specified task.

This paper characterizes the new model of image, as a general framework for image processing, which saves the image into a data structure. For making this model, the image is splitted into sub images. For each sub image, the features such as color, texture, segments, and labels ... are specified then put them in data structure. The features of an image are specified by features of it's sub images. These features are inserted into the data structure in a regular and ordered manner, which presents the features of the image and its sub images hierarchically. Data structures such as binary tree, quad tree, etc. can be used to model them. Each node of data structure will represent features of the image or sub images.

In this paper, we have used a binary tree for data oriented modeling of image. This model uses average and difference features for representing the image or sub images. The

features of original image will be stored in the root of the tree. Each node of tree has two children and stores two features, average color, A, and difference color, D, of it's sub images. For creating this tree, original image is splitted into two sub images with equal size. We do this recursively for each sub image until it turns to a pixel. Thus, Pixels are the leaves of the tree. For each leaf, Color of the pixel is the same for D and A. An attribute A of a parent is equal to average color of its children and D is identical to color difference of its children. This tree named, Average-Difference-Binary-Tree-of-Image, ADBT, and in this paper, it is stored in an array.

Images that model in ADBT must be two splittable squares Image from order k, which is defined below:

Definition: I is a two splittable squares Image from order k if and only if it's width and height be equal to 2^k such that $k=0, 1, 2, 3 \dots$. Let I be a splittable squares Image from order k thus is denoted by I^{2^k} .

ADBT of I^{2^k} Image, which is stored into an array, has the following properties:

- ADBT is an array, which has $2^{2 \times k + 1}$ cells.
- Let i be the Index of one node. Then index of the left child of it is $2 \times i + 1$ and the right child of it is $2 \times i + 2$.
- Leaves of tree are pixels, and they are in $(2k)$ 'th level (depth) and those are stored into tree from $2^k - 1$ index throw $2^{k+1} - 2$ index.
- The nodes of j'th level (depth) are stored into tree from $2^j - 1$ index throw $2^{j+1} - 2$ index.
- Let i be the index of a node in ADBT; the level number (depth) of i is equal to $\lfloor \log_2 i \rfloor$ such that $\lfloor x \rfloor$ is the integral part of x.

To illustrate the concept, Fig. 1 shows an I^{2^1} image, which has four pixels, labeled A, B, C and D respectively.

A	B
C	D

Fig. 1 An I^{2^1} Image

Fig. 2 shows ADBT of Fig. 1. This ADBT has three levels, numbered as 0, 1 and 2. The features of pixels are stored in the leaves. Features of F are obtained by combining features of A and C. In the same way, features of G are obtained from B and D. finally features of the entire image, E, are obtained by combining features of F and G. By putting F and G together, we can achieve smooth of original image as is shown in Fig. 3.

Fig. 4 shows the array, which stores the ADBT of Fig. 2.

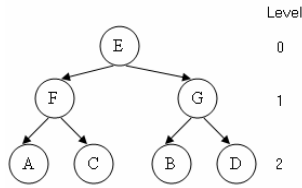


Fig. 2 ADBT of an I^{2^1} image



Fig. 3 Smoothed of original image

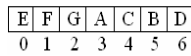


Fig. 4 ADBT of Fig. 2 stored in an array

The following algorithm makes ADBT of I^{2^k} . It should be called as *CreatingADBT* - $I^{2^k}(0, I^{2^k})$.

Algorithm of creating ADBT:

CreatingADBT - $I^{2^k}(\text{index}, \text{image})$

If $\log_2 \text{index} = 2k$ then

Cp = correspond pixel in image

ADBT [index].A = Cp.color

ADBT [index].D = Cp.color

Return

End if

If index is even then

Split image into two equal sub images vertically

FirstSubImage = LeftSubImage

SecondSubImage = RightSubImage

Else

Split image into two equal sub images horizontally

FirstSubImage = TopSubImage

SecondSubImage = BottomSubImage

End if

CreatingADBT - $I^{2^k}(2 \times \text{index} + 1, \text{FirstSub Image})$

CreatingADBT - $I^{2^k}(2 \times \text{index} + 2, \text{SecondSub Image})$

ADBT [index].A = (ADBT [2 × Index + 1].A + ADBT [2 × Index + 2].A) / 2

ADBT [index].D = ADBT [2 × Index + 1].A - ADBT [2 × Index + 2].A

Return

For ease of understanding, mentioned steps are described top to down; but for enhancing speed of making ADBT, it will be created down to top. It means at first semi-peano scanning of image [3] is performed to produce a one-dimensional vector. The difference with original peano scanning is that in our model pixels are ordered top to down, left to right. After that leaves will be initialized by values of produced vector. Therefore, count of leaves is equal to count of pixels of image. Each leaf is corresponded to a pixel of original image and its

A and D are same as the pixel's color. For each non-leaf node, A is equal to average of it's children's A and D is equal to difference of it's children's A.

IV. APPLICATIONS

The ADBT, introduced in the previous section, is a data oriented and applied model of image, which can be applied to many image-processing tasks with desired precision and corresponding speed. i.e. going from down to top, in ADBT's depth, resolution decreases and thus corresponding time for doing tasks decreases too.

If in an application, precision is an unimportant factor, then the required tasks should be done in lower depth for increasing speed of process. Otherwise, those should be done in higher depth for increasing precision. By using this model of image, tasks such as measuring similarity, segmentation, clustering etc could be done with desired precision and corresponding speed.

We give the following definition to outline ADBT's applications in image processing field:

Definition:

The function $U_\alpha(C_1, C_2)$ verifies whether two colors are similar or not; which α is a threshold with selective value. Two colors are similar if and only if Euclidian distance of two colors in RGB color space is less than α .

$$U_\alpha(C_1, C_2) = \begin{cases} 1 & |C_1.R - C_2.R| + |C_1.G - C_2.G| + |C_1.B - C_2.B| < \alpha \\ 0 & \text{otherwise} \end{cases}$$

A. Measuring Similarity of Images

For measuring similarity of two images, we define similarity ratio factor at level L , SR_L , defined as follow:

Definition:

Let $I_m^{2^k}$ and $I_n^{2^k}$ be the two splittable squares image from order k . Let $ADBT_n$ and $ADBT_m$ be corresponding models. SR_L is similarity ratio of $ADBT_n$ and $ADBT_m$ at level L and is calculated by following equation:

$$SR_L = \frac{\sum_{i=2^{L-1}}^{2^{L+1}-2} U_\alpha(ADBT_m(i).A, ADBT_n(i).A)}{2^L}$$

where L indicates level number which ensures the desired precision and corresponding speed. Resulting SR_L is a number in $[0, 1]$ which, low values indicate less similarity and high values indicate more similarity.

B. Edge Detection

Detecting edges is an important task in image processing field. Each node of presented ADBT stores two features, A

and D. D can be used for detecting edges. Suppose β be a selective threshold value thus, If D were greater than β , then midline of its children represents an edge. This line is a line that divides parent node into two sub images with equal size.

If node is located in even depth (Level), then it represents a vertical edge, otherwise a horizontal edge. We can obtain all edges by combining odd level detected edges with even level detected ones.

The Following algorithm detects all of edges by combining level L detected edges with level L+1 ones, suppose L be odd:

```

For i =  $2^{L-1}$  to  $2^{L+1} - 2$ 
  If |ADBT (i).D| >  $\beta$  then
    Detected edge is a line that divides corresponding
    rectangle of ADBT (i) into two equal sub images
    horizontally.
  End if
Next
For i =  $2^{(L+1)} - 1$  to  $2^{(L+1)+1} - 2$ 
  If |ADBT (i).D| >  $\beta$  then
    Detected edge is a line that divides corresponding
    rectangle of ADBT (i) into two equal sub images
    vertically.
  End if
Next

```

C. Clustering

Image clustering is a very challenging problem due to lack of effective representation. A specific region-based tree representation of image incorporating with adaptive processing of the tree has been introduced for doing clustering [1]. Nevertheless, Mentioned approach, employs the EdgeFlow method [8] for segmenting the original images before performing classification. Therefore, it is not a comprehensive representation of image. On the Contrary, presented ADBT has potentials for doing most of image-processing related tasks independently.

For clustering, the main tool is measuring similarity of two images. By using ADBT, we can do this by calculating SR_L , which is defined in section 4.b. By using SR_L in the desired level as a meter along with common clustering algorithms we can do image clustering.

V. CONCLUSION

In this paper, a data oriented model for image is introduced which models image as a tree. Each node of this tree represents the feature of image or sub image. By using the presented model in this paper, tasks as measuring similarity ratio, segmentation, clustering ... can be done *with desired precision and corresponding speed*.

Let n be number of image pixels. Thus to store them we need n memory locations, like BMP format. Corresponding ADBT has $2n-1$ nodes thus we need $2n-1$ memory locations to store them. We can exclude last level of ADBT in many applications, so total number of required nodes will be $n-1$,

which is approximately equal to the number of pixels of the original image, furthermore we can retrieve excluded nodes by combining A and D of their parents.

On the other hand ADBT of I^{2^k} represents 2^k images, smoothed hierarchically, which can be used to do tasks with desired precision and speed.

REFERENCES

- [1] Zhiyong Wang, Dagan Feng, and Zheru Chi, "Region-Based Binary Tree Representation For Image Classification", IEEE International Conference on Neural Networks & Signal Processing, Nanjing, China, 2003.
- [2] Xiaolin Wu, "Image Coding by Adaptive Tree-Structured Segmentation", IEEE Transactions on Information Theory, VOL. 38, NO. 6, 1992.
- [3] G.S.Seetharaman, B.Zavidovique, "Image Processing in a Tree of Peano Coded Images", Computer Architecture for Machine Perception, 1997.CAMP'97. Proceeding Fourth IEEE International Workshop on.
- [4] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high-compression image coding", IEEE Transaction Circuits Syst., vol. CAS-34, no. 11, pp. 1306-1336, Nov 1987.
- [5] R. Leonardi and M Kunt, "Adaptive split-and-merge for image analysis and coding", Proc. SPIE, vol. 594, 1985.
- [6] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video", in ICASSP Proc., May 1991, pp. 2661-2664.
- [7] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation", IEEE Trans. Signal Processing, vol. SP-40, pp.2040-2060, Aug. 1992.
- [8] W. Ma and B. Manjunath, "Edgeflow: a technique for boundary detection and image segmentation", IEEE Trans, on Image Processing, Aug. 2000.
- [9] A.Habibizad Navin, M.N.Fesharaki, M.Mirnia, M.Teshnelab and E.Shahamatnia, "Data Oriented Modeling of Uniform Random Variable: Applied Approach", XXI International Conference on Computer, Electrical, and Systems Science, and Engineering, Vienna, Austria, 2007.
- [10] A.Habibizad Navin, M.N. Fesharaki, M.Teshnelab and E.Shahamatnia. "Fuzzy Based Problem-Solution Data Structure as a Data Oriented Model for ABS Controlling", International Journal of Applied Mathematics and Computer Sciences Vol. 4 No. 2 2007 ISSN 1305-5313.
- [11] http://en.wikipedia.org/wiki/Windows_bitmap
- [12] <http://en.wikipedia.org/wiki/GIF>
- [13] <http://en.wikipedia.org/wiki/JPEG>