Water-Filling: A Novel Way for Image Structural Feature Extraction

Xiang Sean Zhou Yong Rui Thomas S. Huang
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana Champaign, Urbana, IL 61801, USA
{xzhou2, yrui, huang}@ifp.uiuc.edu

Abstract

The performance of a content based image retrieval (CBIR) system is inherently constrained by the features adopted to represent the images in the database. In this paper, a new approach is proposed for image feature extraction based on edge maps. The feature vector with multiple feature components is computed through a "Water-Filling Algorithm" applied on the edge map of the original image. The idea of this algorithm is to obtain measures of the edge length and complexity by graph traverse. The new feature is more generally applicable than texture or shape features. We call this structure feature. Experiments show that the new feature is capable of catching salient edge/structure information in the images. An experimental retrieval system utilizing the proposed new features yields better results in retrieving city/building images than some global texture features (Wavelet moments). The new feature is ideal for images with clear edge structure. After combining the new features with other features in a relevance feedback framework, satisfactory retrieval results are observed.

1. Introduction

Color, texture, and shape are the most frequently referred "visual contents" in CBIR systems [1]. In terms of HSV color model, every pixel in an image can be mapped to a point in the HSV space. In the coordinates H (Hue) and S (Saturation) lies the *chrominance* information, while V (Value) indicates the strength of *illuminance*.

Color feature captures the *chrominance* information in the image. Both texture and shape represent the *illuminance* information. Texture features (e.g. co-occurrence features [2] and Wavelet based features [3]) and shape features (e.g. Fourier descriptors [4] and moment invariants [5]) have been applied extensively in image retrieval systems. However, texture features are effective only for uniform texture images or regions, and shape features require good segmentation and are only effective for simple and clean images.

Real-world images (e.g., natural scenes) are however with noise, changing backgrounds, or object occlusions, etc. The texture and shape features are therefore far from adequate and are limited to deal with only a small fraction of the real-world images. We therefore propose a new set of features capturing more information of the image illuminance, namely, *structure*. Structure is a feature *in-between texture and shape*. It is more general than texture or shape in that it requires neither uniform texture region nor a closed shape contour.

In this paper we explore extraction of structure features from an edge map using our proposed Water-Filling algorithm. Edge map, as a binary image, is obtained by gradient operation followed by thinning. Instead of looking for limited shape information, our proposed algorithm try to answer a more general question of "how to represent the information embedded in the edge map?" The new features extracted from this algorithm are proven to be effective in a retrieval system, especially when combined with other features.

The rest of the paper is organized as follows. In Section 2 we review the related works and existing techniques. In Section 3 we introduce and describe in detail the WF algorithm. Feature extraction based on the new method is presented in Section 4. Section 5 gives the experimental results over real-world images. Conclusions are drawn in Section 6.

2. Background

Shape and boundary feature extraction requires edge linking as the preprocessing step. Without edge linking, "shape" is out of the question; and features such as "edge length", "edge smoothness", or "edge direction" are also hardly meaningful. However, edge linking from an edge map is not trivial at all if we realize that the edge map can be seen as general graphs. A heuristic graph search is not globally optimum thus runs the risk of losing vital information [6]. Whereas a globally optimal method such as dynamic programming (DP) or graph theoretic technique may be computationally too expensive (in some

cases it is NP complete). Furthermore, for a general graph the starting and ending points as well as stages are not well defined as required by DP. Also only extracting the optimal path can loss structure information.

3. The Water-Filling (WF) Algorithm

To address the difficulties encountered in Section 2, we propose an algorithm to extract features from the edge map *directly* without edge linking or shape representation. The idea is to look for measures for the *edge length* and *edge structure and complexity* using a simple and effective graph traverse algorithm.

In a binary edge map, two edge points are *4-connected* (or *4-neighbors*) if the difference in one of their coordinates (x or y) is 0 while in the other, 1. The *connectivity* can be defined in a recursive way: for edge point p, q, and r, p is *connected* to q if p is *4-connected* to q, or p is *4-connected* to r and r is *connected* to q. *8-connectivity* and *m-connectivity* are similarly defined [7].

The algorithm can be regarded as a simulation of "flooding of connected canal systems (connected edges)", hence the name of our algorithm. The assumptions include: i) we have unlimited water supply at the starting point; ii) water flows at a constant speed in all directions; iii) water front will stop only at a dead-end, or at a point where all possible directions have been filled.

As water fills the canals (edges), important statistics are recorded such as maximum filling time; the number of points where the water front forks; and the maximum accumulated fork numbers, and various histograms etc.

The WF algorithm for *4-connectivity* can be described as follows (for simplicity, when we say "*4-neighbors"* we mean the **edge pixels** that are *4-neighbors* of the current pixel, since we only deal with edges):

1). Initialization: mark all edge pixels as "unfilled", WaterFronts={ }, MaxFillingTime= 1, MaxForkCount =1, MaxWaterAmount=0

2). For every "unfilled" edge pixel *p* in the edge map:
   i. Mark it as "filled", FillingTime=1, ForkCount=1, WaterAmount=1, WaterFronts={*p*}, and
   ii. For every pixel *q* not marked as "dead-end" in the set WaterFronts:
      - **If** *q* has *m* "unfilled" *4-neighbors (m>0)*, put them into WaterFronts and mark them as "filled", FillingTime++, ForkCount += *m* – 1, WaterAmount += *m*; and remove *q* from WaterFronts;
      - **else**, mark *q* as "dead-end".
   iii. According to the current values in FillingTime, ForkCount, and WaterAmount:
      - Update MaxFillingTime, MaxForkCount, and MaxWaterAmount
      - Update FillingTime Histogram, ForkCount Histogram, and WaterAmount Histogram.

4. Feature Extraction

Possible features that could be extracted from the above algorithm include:

1). (MFT&FC) MaxFillingTime and the associated ForkCount

MaxFillingTime is the time to fill the "longest" set of edges. The associated ForkCount is the number of forks in this set. These are features most probably associated with a salient object in the image. The MaxFillingTime conveys some measure of the edge length of this object (maximum variation due to the choice of starting point is 50%), while the associated ForkCount gives measure of complexity of the edges (i.e. complexity of the structure of the object);

2). (MFC&FT) MaxForkCount and the associated FillingTime

These are also features most probably associated with a salient object in the image. This object may or may NOT be the same object as the previous one.

3). (FTH&FC) FillingTime Histogram and the associated averaged ForkCount within each bin

This is a global feature on all sets of *connected* edges in the edge map. It represents the edge map by the distribution of edge "length". Noise or changing background with short edges may only affect part of the histogram, leaving the portion depicting the salient objects unchanged. Thus by proper weighting of the components (e.g. by relevance feedback), we could achieve robust retrieval.

4). (FCH&FT) ForkCount Histogram and the associated averaged FillingTime within each bin

This is also a global feature with multiple components. It represents the edge map by the distribution in edge complexity. In case we are not sure about which component is more important, the relevance feedback from the user can guide us to choose the proper weights.

The above features are translation and rotation-invariant. With a normalization factor applied to each component, they are also scaling-invariant if the change in image size is within reasonable range (50%) so that the structure in the image is preserved. Larger size change may result in the change in the edge map thus require modification of the edge detection algorithm. Experiments show that to achieve scaling-invariant, a linear factor should be applied to FillingTime components whereas a squared factor is suitable for the ForkCount components.

Other features include *MaxWaterAmount* and *WaterAmount Histogram* (depicts the total length of the

connected edges and their distribution throughout the image), *Line-Likeness* and *Directionality*.

Furthermore, we can apply WF algorithm on region maps with only minor modification—regions instead of the edges are filled by water. The regional features include *MaxFillingTime* and *FillingTime Histogram* (roughly depicts the size of the region and their distribution throughout the image). *MaxWaterAmount* and *WaterAmount Histogram* (depicts the areas of the regions and their distribution throughout the image).

5. Experiments and Evaluations

Fig1.1 Random picks from a set of 92 images.

We have tested our new features on six sets, a total of 20,000 images. One data set is from COREL database. Four sets are from MPEG-7 testing data sets. Experiments focus on real world images and "retrieval by example". The edge maps are obtained by Sobel filtering followed by a thinning operation [6]. WF features used in these experiments are ***MFT&FC***(see Section 4 for definition)***, MFC&FT, FTH***(7 bins), and ***FCH***(7 bins)—a total of 18 components.

5.1 City/Building images and Landscapes

Fig 1.2. Top 5 retrieved based on WF features. (The top-left image is the query.)

Fig 1.3. Top 5 retrieved based on WF features. (The top-left image is the query.)

The first experiment is carried out on a set of 92 images from Mpeg-7 test set (Fig 1.1). Fig 1.2 and 1.3 show the retrieval results using as a query a city/building image and a landscape image, respectively.

To evaluate the performance, comparison between texture features, namely Wavelet moments, and WF features is performed for city/building image retrieval. First, 17 out of the 92 images are labeled as buildings by a human subject. Each of them is then used as a query image. The number of correct hits out of the top 10 and top 20 returns is counted into Fig 1.4. One can see that WF features outperform Texture features (Wavelet moments) in most cases. In top 10 returns, WF features give an average of 6.2 correct hits, versus 4.7 by Wavelet moments. In top 20 returns, the figures are 9.0 versus 7.1.

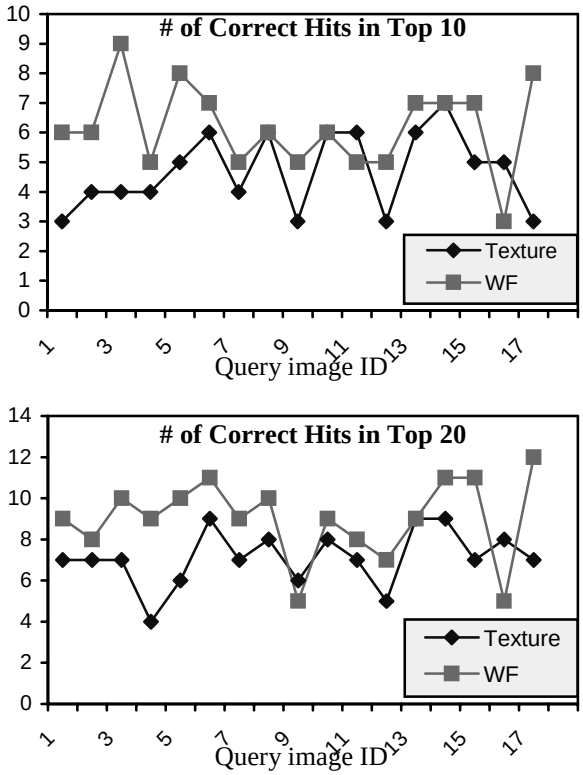

Fig 1.4. Compare Texture features and WF features

Intuitively, windows, doors, and the skylines expose the characteristics of City/building images, which can be captured by the proposed WF features. It is also feasible to extract *edge direction*[9] and *line likeness*, *"L" and "U" junctions*[10], etc., features from WF algorithm, which should improve the performance further.

5.2 Images with clear structure: Birds and Airplanes in the sky

The second retrieval task is performed on the COREL data set of 17,000 images (Fig 2.1). Query images are those with clear edge structure (Fig 2.2 and 2.3). One may notice that retrieval by shape will not be helpful in such cases because from different angles the objects appear dramatically different in shape. But on the other hand, our WF features can mistake birds for airplane, and vice versa, in some cases.

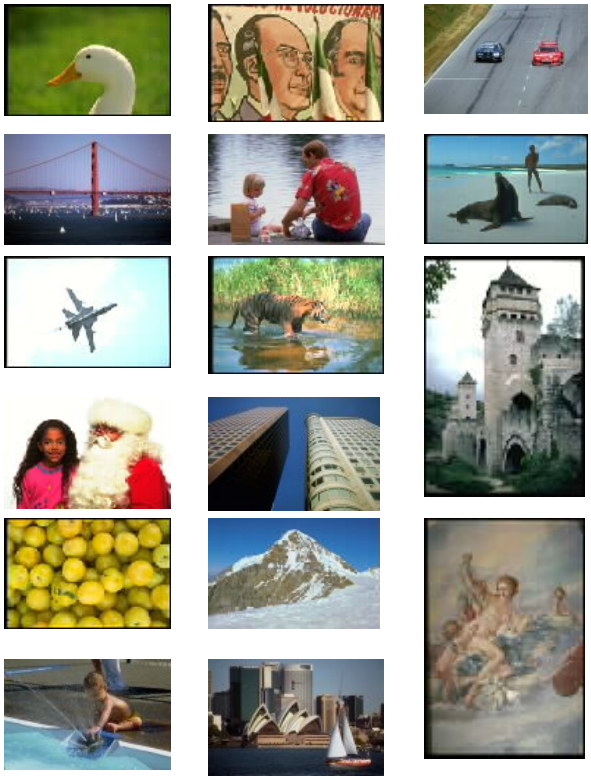

Fig2.1 Random picks from a set of 17,000 images

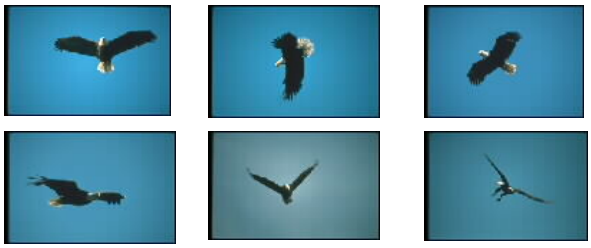

Fig2.2 Top 5 retrieved based on WF features.
(The top-left image is the query image.)

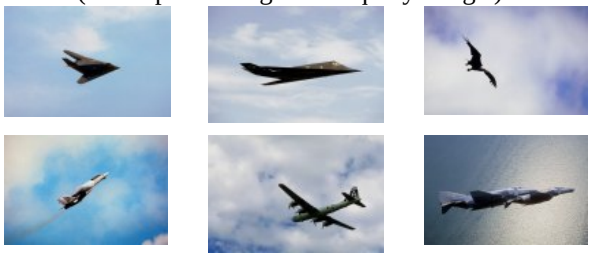

Fig2.3 Top 5 retrieved based on WF features.
(The top-left image is the query image.)

Fig2.4 Top 12 retrieved after Relevance Feedback based on Color, Texture and WF features. (The top-level weights for color, texture and WF are 0.98, 0.38, 1.00)

Fig2.5 Top 12 retrieved after Relevance Feedback based on Color, Texture and WF features. (The top-level weights for color, texture and WF are 0.70, 0.75, 1.00)

5.3 High-level concepts: Horses and Cars

For high-level concepts such as *horses* or *cars*, WF features along will perform very poorly, since it does not contain enough information to characterize the concepts. But if combined with other low-level features and apply relevance feedback technique to incorporate user opinions, then it becomes possible to get reasonably good retrieval results (Fig 2.4 and 2.5). From the feature weights assigned after relevance feedback, one can see that WF features play important roles in the two cases.

One issue worth pointing out is that for example, in the case of retrieving horse images, we are not actually getting "horses" from the system, but rather something like "objects of red and/or white color, of certain size, and of certain edge structure, in a green background of certain texture…", which happen to be horse images in this specific database. Some of the features are not *always* reasonable for the concept, such as the green background in this case. This can be a potential problem for our system as well as many other CBIR systems [9,10,11].

6. Conclusions

In this paper we proposed the WF algorithm to extract edge/structure features directly from edge maps. Based on the experimental results, the new features possess the following characteristics: (1) They can represent information on the edge length, edge connectivity and complexity, as well as the global distribution of such information. (2) They are translation, rotation, and scaling-invariant. (3) They can catch some global information that matches human perception, which may not be carried by texture or shape features. (4) They can be effective on non-uniform real-world images and natural scenes. (5) Since WF features are extracted from the binary edge map, false positive during retrieval is sometimes rather high which indicates the integration with other features is necessary for large databases.

Acknowledgement: This work was supported in part by National Science Foundation Grant CDA 96-24386.

Reference

[1] M. Flickner, et al., "Query by image and video content: The qbic system", IEEE Computers, 1995

[2] R. M. Haralick, et al., "Texture features for image classification", IEEE Trans. On SMC, no.6, 1973

[3] J. R. Smith and Chang, "Transform features for texture classification and discrimination in large image databases", Proc. IEEE ICIP'1995

[4] C. T. Zahn and Roskies, "Fourier descriptors for plane closed curves", IEEE Trans. Computers, 1972

[5] M. K. Hu , "Visual pattern recognition by moment invariants", IRE Trans. on Information Theory, 8, 1962

[6] A. K. Jain, Foundamentals of Digital Image Processing, Prentice Hall , 1989

[7] R. C. Gonzalez and Woods, Digital Image Processing, Addison-Wesley, 1992

[8] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors", IEEE Trans. SMC, 1977

[9] A. Vailaya, A. K. Jain and H. J. Zhang, "On image classification: City Images vs. Landscapes", Pattern Recognition, vol. 31, pp 1921-1936, December, 1998

[10] Q. Iqbal and J. K. Aggarwal, "Applying perceptual grouping to content-based image retrieval: Building images", Proc. IEEE CVPR'99, pp.42-48, 1999

[11] A. L. Ratan, et. al., "A framework for learning query concepts in image classification", Proc. IEEE CVPR'99, pp. 423-429, 1999