International Journal of Science and Technology Volume 2 No. 1, January, 2013

Watershed Transform on Image Segmentation and Data Classification

Parul Vashist, K.Hema
Greater Noida Institute Of Technology,Greater Noida

ABSTRACT

Watershed transform is usually adopted for image segmentation in the area image processing and image analysis The concept of watershed transform is based on a processing simulating the immersion of a landscape in a lake that is dams have to be built to prevent the merging of different catchment basins. In this paper firstly the algorithm of watershed transform are firstly introduced. Two novels methods utilizing watershed transform are proposed. first we propose an effective noise removal method to resolve the problem of small object detection with low contrast and second we propose a methods call "watershed classifier" for data clustering and classification using the watershed transform. Most watershed algorithm are utilized for image data where as the proposed watershed classifier is capable of classifying arbitrary data without prior knowledge.

Keywords: *E-learning, knowledge management, Moodle, SNS, Mahara, learning management system.*

1. INTRODUCTION

A Watershed is a basin-like landform defined by highpoints and ridgelines that descend into lower elevations and stream valleys. Watershed transform can be classified as a region-based image segmentation approach. Image segmentation is the process of isolating objects in the image from the background, i.e., partitioning the image into disjoint regions, such that each region is homogeneous with respect to some property, such as grey value or texture. However, the purpose of watershed transform is not limited in image segmentation, i.e. results generated by watershed transform can be taken as preprocesses for further image analysis.

For example, Parket al. [8] used the boundaries detected by watershed transform to produce active contours (*watersnakes*). Nguyen et al. [9] gave a comprehensive analysis about"*watersnakes*". Another example is a texture measurement technique proposed by Blaffert et al. [10] based on Laplace integral for watershed segmentation. Chien et al. [16] proposed a predictive watershed for fast video segmentation. In Chien's method, watershed lines are updated without searching watersheds in every video frame, which can accelerate the process.

The idea of watershed transform is straightforward by the intuition from geography. Imagine a landscape which is immersed in a lake, catchment basins will be filled up with water starting at each local minimum. After the water level has reached the highest peak in the landscape, all catchment basins are divided by dams, which are called watershed lines. Watershed lines, or briefly speaking watersheds, can divide all the regions (catchment basins) where water will flow. The watersheds may be obtained in two ways, either the boundaries of segmented regions, or the complement of segmented regions. Beucher [1] categorized watershed algorithms into two groups. The algorithms in the first group like immersion algorithm [2] simulate the flooding process. The immersion algorithm [2] is one of the most famous watershed segmentation algorithms. It offers an efficient way to extract watershed lines by simulating the immersion process on gradient images. The second group of watershed algorithms aims at direct detection of watershed lines.

1.1 Watershed Transform (Local Minima)

We can define watershed transform on g. Let $M = \{M_1, M_2, M_3,\ldots.M_k\}$ be the set of all local minima of g . Watershed transform is a labeling process, which maps every point on D to a label domain W, where $W = \{1,2,...,k,u\}$, u is a constant differing from 1,2,...,k

An algorithmic watershed transform is defined as

(1) if $p = m_i$, Wshed (p) = i ,
(2) if dp(p) is undefined, Wshed (p) = u ,
(3) if $p \neq m_i$, Wshed(p) =Wshed(dp(p)), for a selected downhill path dp(p) .

The labeled regions L_1 ,L_2 ,...,L_k ,L_u are then generated, where L {p ЄD |Wshed(p) =i}, i = 1, 2,..., k, u .L_u consists of the points which do not follow their downhill path to any local minima of g. Hence, we also call L_u as "unlabeled region" or "undefined region". This is basically due to the plateau problem. A plateau is a connected subset of D with constant value of g on it. The points in L_u consist of two types:

(1) Points in some plateau,

(2) Points which follow their downhill path to some plateau.

1.2 Watershed Transform (Immersion)

Vincent and Soille [2] proposed an algorithmic definition of watershed transform by simulating immersion process.

Let S_v= {p €D |g (p) ≤ v}: the threshold set for g at level v,

LIM_v = {p€ D | g (p) =v, g(p) < g(q), ¥ (g) € N_n(p)\{p}: local intensity minima set for g at level v ,

v_{min} ,v_{max} : the minimum and maximum value of g .

Define a recursion with the value of g growing from v_{min} to $v_{max:}$

I_{vmin} = {p € D| g(p)=v_{min} }
I_v = LIM_v Ų IZ_{sv} (I_{v-1}) , v=v_{min} +1 , v_{min}+2,…….v_{max}

The watersheds of g is the complement of I_{vmax} :

Wshed (g) = D/ I_{vmax}.

2. DETECTION OF SMALL OBJECTS WITH LOW CONTRAST

Certain critical problems with respect to small-object detection still remain such as Vehicle tracking &Noise. Conventional image differencing methods usually generate poor results in detecting small objects with low contrast. To resolve this problem, an effective and efficient object detection method to detect small moving targets in video sequences. Davies etal. [18] proposed a method for the detection and tracking of small and low contrast objects in FLIR (Forward Looking Infra Red) image sequences. In the detection process, they used larger wavelet filters on multiple images to prevent many noise variations from being detected. A multiple target Kalman tracker was then employed for the tracking of detected objects. However, their method was only applied on FLIR images. The 2-D Lattice algorithm was devised to deal with the problem of detecting small objects in unknown but correlated background clutter A novel noise removal technique is devised by encoding the pixel and its neighbors according to the noise distribution to effectively remove most of the possible non-object noise. Then, the regions of interest (ROI) are located by thresholding the denoised image. After locating the ROI, a segmentation algorithm incorporating region matching technique is employed on the ROI to extract better object contours. As to the problem of small object detection, it is difficult to find the complete object contour. The watershed lines can effectively divide individual catchment basins in a gradient image, and generate closed contours for each region in the original image.

Watershed-based segmentation algorithm, adopt the MGD criterion to generate watershed lines and replace the LIM with the drowning threshold criteria. Experimental results demonstrate that the proposed algorithm performs well in extracting contours of small objects. Moreover, the execution speed of the proposed algorithm is extremely fast which will make real-time applications possible.

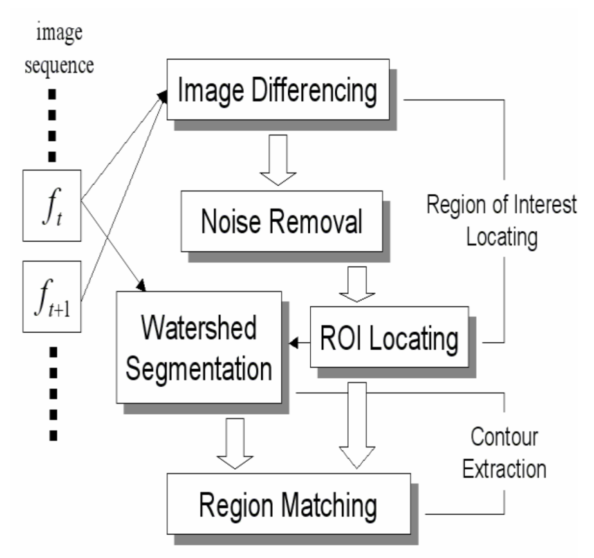


Figure 1.The scheme for moving object detection.

2.1 Noise Removal Locating

Low signal to noise ratio (SNR) is a serious problem for the detection of small objects with low contrast in image sequences. Under this circumstance, the number of noise pixels could be much more than that of target pixels. Hence, histogram-based techniques adopted in conventional image differencing methods are inappropriate to resolve the problem. Median-filter-based methods are also popular approaches for noise reduction in image processing. However, they will eliminate not only noise but also small targets. To remedy this problem, a novel noise removal algorithm is developed to remove the clutter noise by encoding every pixel and its neighbors according to the distribution of noise. Once the noise is removed, the regions of interest (ROI) can thus be accurately located for later contour extraction.

2.2 Region of Interest Locating

The pixels whose difference values are greater than the threshold value T remain to be the candidates for moving objects. However, the contour of the found objects is not clear or complete under the current circumstance. More image processing steps are required to improve the quality. After extracting the object pixels, the center of the object pixels (x, y) is defined as the initial location for ROI determination. The ROI centered at (x, y) can thus be determined by a window of size w× h for later contour extraction process. a window of size $w \times h$ for later contour extraction process.

2.3 Contour Extraction Using Watershed-Based Segmentation and Region Matching

Beucher [1] categorized watershed algorithms into two groups. The first group contains algorithms which simulate the flooding process, like the immersion algorithm [2]. The second group of watershed algorithms is aiming at direct detection of watershed lines.

The proposed watershed algorithm belongs to the second group with the merits of simplicity without losing efficiency. Hernandez [3] proposed a watershed-based algorithm for image segmentation by using the properties of **L**ocal **I**ntensity **M**inima (LIM) and **M**orphological **G**radient **D**irection (MGD) both pre-filtering and post-filtering are not suitable for extracting contours of small objects due to the fact that the contours will not remain intact if both two techniques (pre-filtering and post-filtering) are used. Therefore, it is important to resolve the over-segmentation problem.

The watershed algorithm is performed on the gradient of the original image to extract the object contour, not on the difference image. Let us consider a pixel (x, y) in a gray-valued image. The gradient magnitude g for pixel (x, y) is defined as the square-sum-root of its horizontal and vertical gradient vectors. If the gradient magnitude of a pixel is lower than the drowning threshold t, this pixel is labeled as the pixel belonging to an initial region. After all pixels have been\examined, all labeled pixels that are connected are grouped together and assigned with the same label.

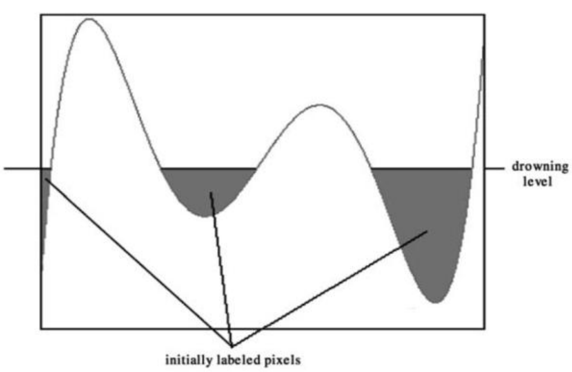


Figure 2: The drowning threshold and the initially labeled regions

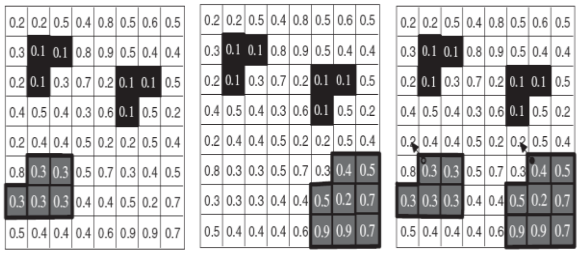
The watershed-based segmentation algorithm can be formally stated as follows:

Algorithm *Watershed-based Segmentation*

```
for each pixel P_i in gradient image f
    do if gradient[P_i] < drowning_threshold
            then label[P_i] ← newlabel
RelabelAllConnectedLabeledPixels()
for each unlabeled pixel P_i in gradient image f
    do P ← P_i
        stack.clear()
        while notlabeled(P)
            do stack.push(P)
                P ← GradientDirection[P_i]
        ReachedLabel ← label[P]
        while stack.nonempty()
            do P ← stack.pop()
                label[P] ← ReachedLabel
```

In performing the proposed watershed-based segmentation algorithm, two common problems are identified which cause the unlabeled regions. An unlabeled region is formed by those unlabeled pixels which cannot “flow” to any existing labeled region. The occurrence of the unlabeled regions will usually affect the results of the contour extraction, which is especially prominent for small objects. Firstly, since the small regions contain few levels of gradien values, the plateau problem will sometimes occur. Fig (a) illustrates an example, in which an unlabeled region resulting from the plateau problem is demonstrated. In this figure, the plateau region is represented by the gray area. Since pixels in the plateau regions contain the same gradient magnitude, it is possible that these pixels will reach different regions. This phenomenon is not reasonable for most cases. A simple way to remedy this problem is to group these pixels into a“larger pixel” before performing the watershed-based segmentation algorithm. In this way, the pixels in the plateau region will always point to the neighbor with the lowest gradient value.

The second problem is that the unlabeled regions contain the local minimum of the gradient magnitude that is greater than the drowning threshold see Fig. (b). The directions of all pixels surrounding the unlabeled regions will point to the local minimum inwardly. Hence, the pixels in this kind of unlabeled region will not reach any initial region. Again, we can resolve this problem with a similar method. The unlabeled region can be treated as a “larger pixel and assigned to the identity of the neighbor with the lowest gradient value. Fig.(c) illustrates the assignment of the “larger pixel” for resolving both the plateau and the unlabeled region problems. By doing so, the occurrence of unlabeled regions that hinders the watershed-based segmentation algorithm can be solved by the proposed idea.



(a) (b) (c)

Figure 3. Examples of unlabeled-region problems.

(a) An example illustrating the plateau problem. The region in the gray area is a plateau (the drowning threshold t = 0.1).

(b) The region highlighted by the gray area is an unlabeled region. Every pixel inside the unlabeled region cannot follow its gradient direction to reach a labeled region.

(c) The plateau and unlabeled regions as the so-called “larger pixels’’. We also found out that watershed segmentation is sensitive to local contrast variations even if they are relatively very low in the global view. Moreover, watershed-based segmentation always produces closed regions. Hence, we can usually obtain complete and noise-free contours of the objects.

2.4 Region Matching

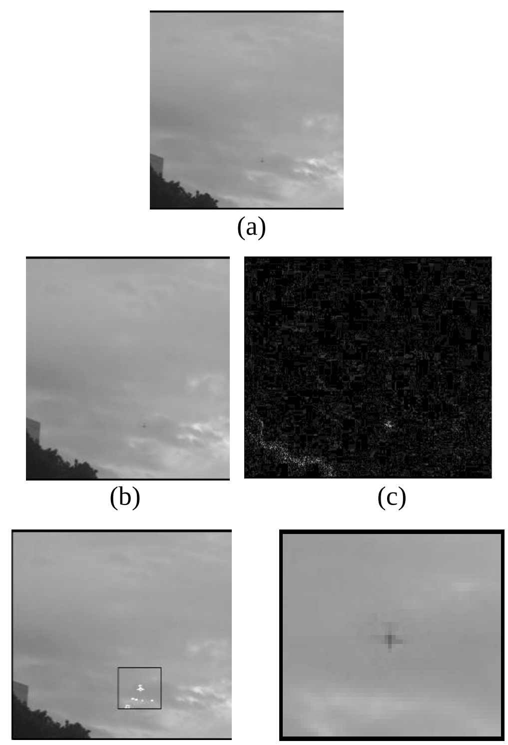
Since the watershed-based segmentation algorithm performs directly on the gradient image, it might generate pseudo-objects resulting from background clutter. To discriminate pseudo-objects from real objects, we can examine the results generated by the watershed-based segmentation algorithm and the results in the difference image after noise removal. If an object is a real one, it must have a certain degree of correspondence among these two results. According to the above criterion, all regions extracted by the proposed watershed-based segmentation algorithm are examined. The regions containing 50% or more non-zeros pixels in image D are identified as the real objects. This step is termed as “region matching”.

2.5 Davies’ Method

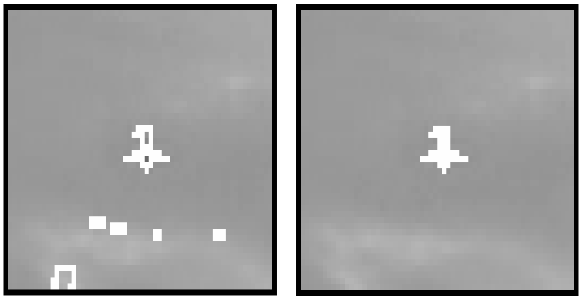
The effectiveness of our proposed method in detecting small objects with low contrast is compared with Davies’ method [18]. Before conducting the comparison, a brief review of avies’ method. In Davies’ method, Daubechies wavelet filters [22] are applied temporally at a fixed ixel location to combine multiple images. The results of temporal filtering are treated as positive detection of targets. The Daubechies wavelet filters of lengths 4, 6 and 8 are used. These filters then provide the generalization of image differencing to combine 4, 6 and 8 images in a sequence together. The high pass nature of the filter then detects the passage of a grey-level boundary across each pixel and thereby highlights the outline of the object.

2.6 Experimental Results

Our targets are aircrafts, and a video camera with 30 frames/s was adopted to capture image sequences containing small aircrafts. All images are 8-bit gray-scale and 325× 240 in size. Shown in Fig. 3.6 is an example illustrating the whole object detection process. Fig. 3.6(a) and (b) are two consecutive images containing a small flying aircraft, and the difference image is shown in Fig. 3.6(c). Fig. 3.6(d) is the result with a drowning threshold t = 0.05. Shown in Fig. 3.6(e) is the enlarged ROI. In the detected region, there are still some small blobs located in the vicinity of the detected aircraft, as can be perceived in the enlarged image of Fig. 3.6(f). We caneliminate these small non-object blobs by overlapping the detected region with the same region in the difference image and the result is shown in Fig. 3.6(g). Figs. 3.7 and 3.8 are another two examples containing aircrafts in different environments. Fig. 3.7(a) shows an image with an aircraft in a cloudy sky and the detection result is illustrated in Fig. 3.7(b). Fig. 3.8(a) shows an image with weaving trees beneath the aircraft and the detection result is illustrated in Fig. 3.8(b). Experimental results reveal that the proposed approach can successfully detect aircrafts in various environments.



(a) (b) (c) (d) (e)



(f) (g)

Figure 4. An example illustrating a detection result

(a) and (b) are two consecutive aircraft images; (c) the difference image of (a) and (b); (d) the detection result where t = 0.05 ; (e) the enlarged ROI; (f) the enlarged detection result; (g) the extracted object contour.

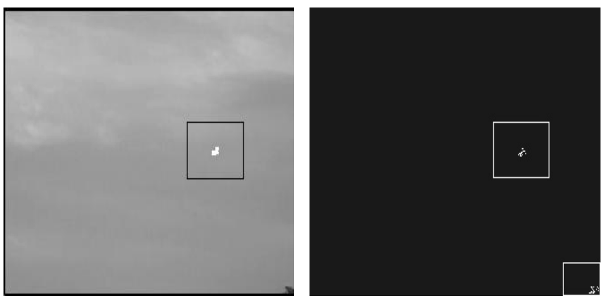


Figure 5. An example illustrating the comparison of small object detection between the proposed method and Davies' method

(a) the detection result generated by the proposed method;

(b) the detection result generated by Davies' method. A false positive detection is produced by Davies' method as depicted in the right-bottom corner of (b).

Davies' method is also applied on the same test samples for comparison. Apparently, the results generated by the temporal filtering proposed by Davies always contain more or less noise. Since they do not consider the factor influenced by noise, the noise will be treated as a positive detection result. To conduct a more reasonable comparison, isolated noise has to be removed first and a simple thresholding procedure has to be employed to remove weak response noise. If there is any noise that does not obviously belong to an object region, it will be regarded as a false positive detection. On the other hand, if the number of pixels in the detected objects does not exceed 50%, they will be regarded as false negative detections. The experimental data were evaluated in this manner among 3,111 test images. In total, there are 2,772 correct detections and the accuracy rate is 89.1% as tabulated in Table

Table: Comparison between the proposed method and Davies' method for the accuracy rates

Sequence	No. of images	Correct	Incorrect	Accuracy rates (%)	Davies (%)
1	251	245	6	**97.61**	95.62
2	71	70	1	**98.59**	92.96
3	179	170	9	**94.97**	90.50
4	305	288	17	**94.43**	81.64
5	144	142	2	**98.61**	97.22
6	125	123	2	**98.40**	88.00
7	179	171	8	**95.53**	88.27
8	377	365	12	**96.82**	84.88
9	233	214	19	**91.85**	86.70
10	287	266	21	**92.68**	84.32
11	215	209	6	**97.21**	93.02
12	99	99	0	**100.0**	95.96
13	144	140	4	**97.22**	90.28
14	179	174	5	**97.21**	94.97
15	144	138	6	**95.83**	92.36
16	179	165	14	**92.18**	86.59
Total	3111	2979	132	**95.76**	89.10

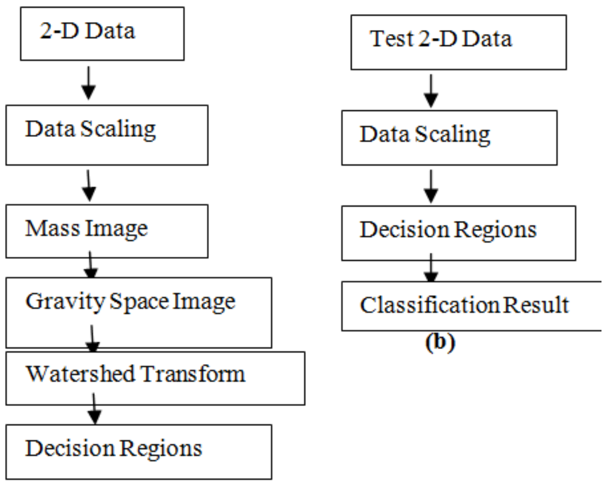
3. DATA CLASSIFICATION USING WATERSHED TRANSFORM

Data clustering and data classification are two basic processes in data analysis and had been widely applied in various research areas, such as pattern recognition and data mining. To classify an incoming data sample, the process of data classification is usually performed after the

process of data clustering. Abundant of data clustering and data classification methods were proposed in the past decades. Typical data clustering methods, such as *max-min*, *K-means*, *Isodata* [23] and *fuzzy c-mean* [24] algorithms, are able to separate data into clusters in different manners. The clustering algorithms only form various clusters without the classification functionality.

On the other hand, the purpose of data classification is to generate decision functions from the various clusters on which the classification is based. The generated decision functions can thereby be used in classifying new incoming data. For example, *Bayes classifier* [23][25], *neural networks* [23][26], *HMM* [27-31], and *SVM* [32-35] are all well-known data classifiers. Generally, data classifiers are divided into two categories: supervised classifier and unsupervised classifier. Supervised classifier classifies data with the need of prior knowledge (such as the belonging class of each training data point), whereas unsupervised classifier can classify data without the need of prior knowledge. A novel unsupervised 2-D data classification method utilizing watershed transform is proposed. For a set of unsupervised 2-D data, our proposed method is capable of automatically generating the decision regions, which are experimentally proven to be optimal in classifying incoming data due to the well design of gravity-space image and the morphological behavior of watershed transform. The task of classification can be accomplished simultaneously via the generated decision regions without needing utilization of decision theory. The proposed method consists of two main stages. In the first stage, the gravity-space image is generated from the set of data. Watershed transform is then manipulated on the gravity-space image to delimit the decision regions in the second stage. In the proposed approach, the input 2- Next, a gravity-space image is generated from the mass image. Lastly, watershed transform is employed on the gravity-space image to extract the decision regions.

3.1 Flowcharts of the Proposed Method



(a)

(a) The procedure for extracting the decision regions using watershed transform

(b) The procedure for classifying incoming 2-D data using decision regions.

The regions extracted by watershed transform are treated as the decision regions for classification purpose. Smet et al. [4] proposed a rain-falling watershed algorithm to control the number of catchment basins by setting up the drowning threshold criterion, which is also adopted in our work on the detection of small objects with low contrast.

3.2 Data Scaling and Mass Image Generation

In our work, data scaling serves as the first step in generating the decision regions from a set of 2-D data. The goal of data scaling is to transform the data into the scaled data being bounded in a fix-sized space. The benefit of data scaling is to reduce the computational complexity to facilitate the later processes, especially when the original data spreads in a very wide range. A mass function can then be defined by manipulating the scaled data to generate a mass image.

3.3 Data Scaling

Let D be a countable set of 2-D data and the i_{th} *elements* of D is denoted by (x_i, yi). The upper bound values of x_i and y_i are given by $u_x = \max\{x_i\} + \alpha$ and $uy = \max\{y_i\} + \alpha$ respectively. The lower bound values of x_i and y_i are given by $l_x = \min\{x_i\} - \alpha$ and $l_y = \min\{x_i\} - \alpha$ respectively. Here, α is an expansion parameter with $\alpha \geq 1$. A set of scaled data S_D of D is the collection of (s_i, t_i) for each (x_i, y_i) in D, where (s_i, t_i) can be obtained by a scaling mapping f defined as

$$f(x,y) = \left[\frac{a(x-l_x)}{u_x - l_x}, \frac{b(l-l_y)}{u_y - l_y}\right]$$

In the above equation, a and b are two positive integers. By a simple derivation, we can find that (s_i, t_i) will fall in the range of $(0,a)*(0,b)$. The inverse mapping $\sim f$ of f can then be obtained from equation

$$\sim f(s,t) = \left(\frac{s(u_x - l_x)}{a} + l_x, \frac{t(u_y - l_y)}{b} + l_y\right)$$

3.4 Generation of Mass Image

After the scaling process has been operated on the 2-D data, a mass image can be constructed from the scaled data S_D. The mass image M_D is defined as

$$M_D(x, y) = \{(s,t) \mid (s, t) \in S_D, x-1 \leq s < x, y-1 \leq t < y\}$$

where $x, y \in N$, $1 \leq x \leq a$, and $1 \leq y \leq b$. Briefly speaking, $M_D(x, y)$ represents the number of elements of S_D falling in the range of $[x-1, x) \times [y-1, y)$. With the mass image being constructed, a gravity-space image can thereby be generated on which further watershed transform is performed.

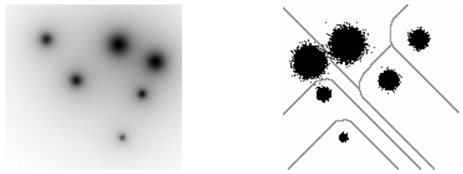
3.5 Gravity-Space Image

The gravity-space image G_D can be directly derived from the mass image M_D. The characteristics of G_D can be observed and summarized as follows:

1. **The values of G_D fall in the range of [0,1] .**
2. **Densely-distributed areas in M_D form low-gradient areas (e.g. catchment basins) in G_D.**
3. **Sparsely-distributed areas in M_D form high-gradient areas (e.g. hillside, plateau) in G_D.**

Due to these inherent characteristics of G_D, the decision regions extracted from gravity-space image will obtain excellent classification result. The performing of watershed transform on gravity-space images will result in the fact that for each of catchment basins where data points are densely distributed, one decision region is formed.

(a) **(b)**



(c) (d)

An example demonstrating the step-by-step results generated by the proposed method. (a) A set of scaled 2-D data, (b) the mass image generated from (a), (c) the gravity-space image generated from (b), (d) the decision regions along with classification map generated by applying watershed transform on (c).

4. WATERSHED TRANSFORM

Beucher[1]categorized watershed algorithms into two groups. The algorithms in the first group like immersion algorithm [2] simulate the flooding process. The second group of watershed algorithms aims at direct detection of watershed lines. In our work, the watershed algorithm that we adopt belongs to the second group with the merits of simplicity without loss of efficiency. Here, the watershed algorithm has been modified to suit our need.

Hernandez et al. [3] proposed a watershed-based algorithm for image segmentation by using the properties of local intensity minima (LIM) and morphological gradient direction (MGD). In our work, the LIM-based approach is inappropriate for generating decision regions because LIM-based approach generates too many small regions (so-called over-segmentation) in many practical cases. Smet et al. [4] proposed a rain-falling watershed algorithm using the drowning threshold to effectively alleviate the phenomenon of over-segmentation. A drowning step is executed before the rain-falling step, and then certain numbers of lakes are created to group all the pixels that lie below a threshold. They indicated that the drowning threshold can be interpreted as an “underground water level”, and is proven to be useful in reducing the over-segmentation phenomenon. However, if the data of a cluster is too sparse to generate an initial group, watershed transform using drowning threshold criterion will fail in extracting a decision region for the cluster.

A plateau problem will sometimes occur in watershed transform process. A continuous area with the same gradient magnitude (or gravity value) is called a plateau. Since the interior pixels of a plateau cannot be directed to any neighbor of them, they will not be labeled with a label of any initial group. In our work, a lower-completion criterion is adopted to solve the plateau problem.

4.1 Data Classification

The procedure of data classification is depicted as follows: For an incoming 2-D data sample (*x*,*y*), calculate*f* (*x*,*y*) formulated in equation (4.1) to obtain a scaled coordinate Then, assign the label of (*x*, *y*) to the class with the label of integer coordinate ([x̃] [ỹ]) in the watershed results and then (*x*, *y*) is classified.

4.2 Supervised Classification

In supervised classification, the initial regions are identified by human intervention instead of using CDP in the unsupervised classification. In the unsupervised version, all pixels in a CDP are grouped and marked with an initial label. In the supervised version, we use markers instead as initial regions for performing watershed transform. Users provide the position, size, shape, and number of the markers. Basing on the initial regions, watershed transform will extract the decision regions as required. A merit of supervised Basing on the initial regions, watershed transform will extract the decision regions as required. A merit of supervised classification is that the attributes (position,size, shape, and number) of markers are assigned by human, whereas the attributes of markers generated by a machine in the unsupervised version might be erroneous.

5. GENERALIZED WATERSHED CLASSIFIER

With the development of watershed classifier for 2-D data, the need of generalized watershed classifier is emerged. The difficulties in generalizing the 2-D watershed classifier to higher imension (more than two features) are various. The most difficult problem could be that the watershed transform is originally a morphological operation on 2-D surface. Important watershed algorithms, such as immersion algorithm [2], are basically defined and devised limited to 2-D domain. Fortunately, rain-falling based watershed algorithms [1][3][4] are not natively limited to 2-D domain. algorithms [1][3][4] are not natively limited to 2-D domain. Hence, it is possible to generalize the rain-falling algorithm to high dimensional domains.

5.1 Gravity Values

The calculation of gravity values in the generalized watershed classifier is slightly different from that of 2-D watershed classifier. In 2-D watershed classifier, a gravity-space image is prepared for watershed transform. The gravity value of each point in the gravity-space image is obtained by calculating the summation of the influences from all points in the mass image. The definition of gravity values g_D is given by

$$g_D(x) = \sum_{y \in M_D,\, y \neq x} \underline{M_D(x)} + \gamma M_D(x)\ d(x,y)^2$$

where $d(x, y)$ is the Euclidean distance between data points x and y . Here $\gamma = 1.2$ is an empirical value obtained by experimental setting.

5.2 Discussions

Optical character recognition (OCR) is a traditional and important work in pattern recognition. OCR is usually carried out through two main steps, one is feature extraction and the other is feature classification. In our work, the recognition of Arabic numerals using various fonts is adopted as the sample experiment which is performed by utilizing our generalized data classifier. a generalized watershed classifier (GWC) is devised. The methodology of GWC is based on the design of high dimensional gravity values and cooperating with generalized rain-falling watershed algorithm. Experimental results verify the validity of our proposed generalized watershed classifier.

6. CONCLUSIONS

Watershed transform is a powerful tool for the purpose of image segmentation. However, the purpose of watershed transform is not limited in image segmentation. In this dissertation, two novel methods implemented based on watershed transform are proposed, including small object detection and data classification. For specific applications using watershed transform, customized modification for watershed transform in necessary. As to the problem of small object detection, it is difficult to find the complete object contour. To resolve the problem, an efficient watershed-based algorithm is devised to extract better object contours from the ROI. The drowning level criterion is utilized in a rain-falling watershed algorithm to alleviate the over-segmentation problem in extracting the contours of small objects using watershed transform. However, the drowning level criterion is not suitable to be adopted in our proposed watershed classifier as depicted in chapter 4. If the data in a cluster is too sparse to generate an initial label by the drowning level criterion, watershed transform will fail in generating a decision region for the cluster in our approach. After clarifying the serious drawback of using a global drowning threshold, a rain-falling watershed algorithm is developed instead by introducing a marker called **C**onstant **D**epth **P**ool (CDP). Experimental results show that the utilization of CDPs as markers in watershed transform to perform the classification task generates satisfactory results. In our novel method of detection of small object with low contrast, an efficient noise removal technique is first devised. This technique is operated on the difference image to effectively remove noise. With the bothering noise removed, accurate ROI can then be located. Finally, a watershed-based segmentation algorithm is employed on the ROI to extract the object contours. Experimental results demonstrate that the watershed-based segmentation performs well in extracting the contours of small objects with low contrast. By incorporating the region matching technique, better object contours can be detected. Experimental results indicate that the proposed method is feasible and efficient in detecting the contours of small moving objects.

In our proposed novel method of watershed classifier, watershed transform in firstly employed on the area of arbitrary data besides image data. The proposed method consists of two main stages. In the first stage, the gravity-space image is generated from the set of data. Watershed transform is then manipulated on the gravity-space image to delimit decision regions in the second stage. Conventionally, the task of data classification is performed by applying a clustering algorithm on unsupervised data to form various data clusters first and certain decision function generation algorithm is then operated to generate the decision functions. Lastly, incoming data is classified by using the decision functions based on the decision theory. In our work, the proposed method is capable of automatically generating the decision regions for a set of unsupervised 2-D data, which are experimentally proven to be optimal in classifying incoming data due to the well design of gravity-space image and the morphological behavior of watershed transform. The task of classification can be accomplished simultaneously via the generated decision regions without needing the utilization of decision theory. The proposed method generates excellent classification results without prior knowledge of data. When the amount of data is huge, our method can still provide very fast processing speed comparing with other methods. A supervised version of the proposed method is also devised by using markers as the initial regions for watershed transform to enhance the classification performance. In our work, a preliminary development of generalized watershed classifier (GWC) is devised. The methodology of GWC is based on the design of high dimensional gravity values and cooperating with generalized rain-falling watershed algorithm. Experimental results verify the validity of our proposed generalized watershed classifier.

REFERENCES

[1] Beucher, S.The Watershed Transform Applied to Image Segmentation, *Proceedings of the P.Pfefferkon Conference on Signal and Image Processing in Microscopy and MicroAnalysis*,pp.299-314,September 1991.

[2] Vincent, L., and Soille, P. Watersheds in Digital Spaces:An Efficient Algorithm Based on Immersion simulations, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol.13,no:06,pp.583-598,June 1991.

[3] Hernandez, S.E., and Barner, K.E. Joint region merging criteria for watershed-based image segmentation, *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 108–111, 2000.

[4] Smet, P.D., Luis, R., and Pires, V.P.M. Implementation and analysis of an optimized rainfalling watershed algorithm, in: *Proceedings of the Science and Technology Conference, Image and Video Communications and Processing*, January 2000.

[5] Roerdink, J. B. T. M., and Meijsterm, A. The Watershed Transform: Denitions, 102 Algorithms and Parallelization Strategies, *Fundamenta Informaticae 41*, pp. 187-228, 2001.

[6] Digabel, H., and Lantuejoul, C. Iterative algorithms, *Actes du Second Symposium Europeen d'Analyse Quantitative des Microstructures en Sciences des Materiaux, Biologie et Medecine, Caen, 4-7 October 1977*, J.-L. Chermant, Ed., Riederer Verlag, Stuttgart, pp. 85-99, 1978.

[7] Lantuejoul, C. *La squelettisation et son application aux mesures topologiques des mosaiques polycristallines*. PhD thesis,cole des Mines, Paris, 1978.

[8] Park, J., and Keller, J.M. Snakes on the watershed, *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 23, no. 10, pp. 1201-1205, October 2001.

[9] Nguyen, H.T., Worring, M., and Boomgaard, R.V.D. Watersnakes: energy-driven watershed segmentation, *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 25, no. 3, pp. 330-342, March 2003.

[10] Blaffert, T., Dippel, S., Stahl, M., and Wiemker, R. The Laplace integral for a watershed segmentation, *Proceedings of 2000 International Conference on Image Processing*, vol. 3, pp. 444-447, 2000.

[11] Moga, A. *Parallel watershed algorithms for image segmentation*, PHD Thesis, Tampere University of Technology, Tampere, Finland, February 1997. 103

[12] Moga, A.N., and Gabbouj, M. Parallel maker-based image segmentation with watershed transformation, *Journal of Parallel and Distributed Computing 51*, pp. 27-45, 1998.

[13] Klein, J.C., Lemonnier, F., Gauthier, M., and Peyrard, R. Hardware implementation of the watershed zone algorithm based on a hierarchical queue structure, *Proceedings of IEEE Workshop on Nonlinear Signal and Image processing*, Neos Marmaras, Halkidiki, Greece, I. Pitas, Ed., pp. 859-862, June 1995.

[14] Noguet, D., Merle, A., and Lattard, D. A data dependent architecture based on seeded region growing strategy for advanced morphological operators, *Mathematical Morphology and its Applications to Image and Signal Processing*, P. Maragos, R. W. Shafer, and M. A. Butt, Eds. Kluwer Acad. Publ., Dordrecht, pp. 235-243, 1996.

[15] Kuo, C.J., Odeh, S.F., and Huang, M.C. Image segmentation with improved watershed algorithm and its FPGA implementation, *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 753-756, 2001.

[16] Chien, S.-Y., Huang, Y.-W., and Chen L.-G. Predictive watershed: a fast watershed algorithm for video segmentation, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, issue 5, pp. 453-461, May 2003. 104

[17] Beare, R. A locally constrained watershed transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, issue 7, pp. 1063-1074, July 2006.

[18] Davies, D., Palmer, P., and Mirmehdi, M. Detection and tracking of very small low contrast objects, *Proceedings of the British Machine Vision 9th Conference*, pp. 599–608, September 1998.

[19] Ffrench, P.A., Zeidler, J.R., and Ku, W.H. Enhanced detectability of small objects in correlated clutter using an improved 2-d adaptive lattice algorithm, *IEEE Trans. on Image Process.* 3 (6), pp. 383–397, 1997.

[20] Sonka, M., Hlavac, V., and Boyle, R. *Image Processing, Analysis, and Machine Vision, 2nd ed.*, Brooks/Cole Publishing, pp. 77–82, 1999.

[21] Scheaffer, R.L. *Introduction to Probability and its Applications, 2nd ed.*, The Book Company, pp. 285, 1995.

[22] Daubechies, I. *Ten Lectures on Wavelets*, SIAM, 1992.

[23] Tou, J.T., and Gonzalez, R.C. *Pattern Recognition Principles*, Addison-Wesley Publishing, 1974.

[24] Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated cluster, *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.

[25] Rish, I. An empirical study of the naive Bayes classifier, *IJCAI 2001 Workshop* 105 *on Empirical Methods in Artificial Intelligence*, 2001.

[26] Fukushima, K. Cognitron: A Self-Organizing Multilayered Neural Network, *Biological Cybernetics 20*, pp. 121–136, 1975.

[27] Baum, L.E., and Petrie, T. Statistical inference for probabilistic functions of finite state Markov chains, *Ann. Math. Stat.*, vol. 37, pp. 1554-1563, 1966.

[28] Baum, L.E., and Egon, J. A. An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology, *Bull. Amer. Meteorol. Soc.*, vol. 73, pp. 360-363, 1967.

[29] Baum, L.E., and Sell, G. R. Growth functions for transformations on manifolds, *Pac. /. Math.*, vol. 27, no. 2, pp. 211-227, 1968.

[30] Baum, L.E., Petrie, T., Soules, G., and Weiss, N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Ann. Math. Stat.*, vol. 41, no. 1, pp. 164-171, 1970.

[31] Baum, L.E. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, *Inequalities*, vol. 3, pp. 1-8, 1972.

[32] Vapnik, V. *Statistical Learning Theory*. New York: Wiley, 1998.

[33] Bernhard, S., Christopher, J.C.B., and Alexander, J.S. *Advances in Kernel Methods*, The MIT Press, 1998. 106

[34] Hsu, C.-W., and Lin, C.-J. A simple decomposition method for support vector machines, *Machine Learning 46*, pp. 291-314, 2002.

[35] Hsu, C.-W., and Lin, C.-J. A comparison on methods for multi-class support vector machines, *IEEE Transactions on Neural Networks*, vol. 13, pp. 415-425, 2002.

[36] Lucas, S.M., and Cho, K.T. Fast convolutional OCR with the scanning N-tuple grid, *Proceedings of 8th International Conference on Document Analysis and Recognition*, pp. 799-803, August 2005.

[37] Beucher, S., and Lantuejoul, C. Use of watersheds in contour detection, *Proceedings of International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, September 1979.

[38] Meyer, F., and Beucher, S. Morphological Segmentation, *Journal of Visual Communication and Image Representation*, vol. 1, Academic Press, pp. 21-46, September 1990.

[39] Beucher, S., and Meyer, F. The morphological approach to segmentation: the watershed transformation, *Mathematical Morphology in Image Processing*, E. R. Dougherty, Ed. Marcel Dekker, New York, ch. 12, pp. 433-481, 1993.

[40] Beucher, S. Watershed, hierarchical segmentation and waterfall algorithm, *Mathematical Morphology and its Applications to Image Processing*, J. Serra and 107 P. Soille, Eds. Kluwer Acad. Publ., Dordrecht, pp. 69-76, 1994