IAPR Workshop on CV - Special Hardware and Industrial Applications OCT.12-14, 1988, Tokyo

OBJECT RECOGNITION IN THE COMPLEX TEXTURED BACKGROUND

Yu Xiaohan Yuan Baozong

Research Institute of Information Science
Northern Jiaotong University
Beijing, P.R.CHINA

Abstract:

A new object extraction algorthm is suggested in this paper, which is based on the feature difference between the object surface and the texture background; In essence, the algorithm is the extension of those algorithms which extract objects by analyzing the gray distribution of the image. The new algorithm not only can extract the objects in the texture background with different gray distributions but also with the same gray distribution. If the background is non-stationary (made up of several kinds of textures), our algorithm is still effective.

As a application, we used our algorithm to do object recognition in the textured background. For different natural textured backgrounds and ten polyhedron objects, the classification rate is about 98%.

1. Introduction

Computer vision has attracts more and more attention in recent years. The identification and understanding of 2_D and 3_D objects can be made precisely. The precision has met the needs of practical applications; But most present vision methods can only deal with the images with either white or dark background. Unfortunately, many natural pictures such as remote sense images and SAR images have complex texture backgrounds. These images can not be processed directly by present vision methods. So it is necessary to separate objects from texture backgrounds, and get the object images without any texture backgrounds. Then vision methods can be used to recognize or interpret the images.

There are two kinds of object extraction algorithms. The first kind is based on the assumption that the average gray level of object surface is different from that of the background[1]. So a proper threshold can be selected in the gray histogram and be used to separate the object region from the background region. This kind of algorithm is usually easy to implement[1], but has limited applications. The second kind is based on the modeling of the texture background. The principle is that the object region and the background region have different stochastic models. These algorithm are more general, but more complex than the first kind. The experimental results in [2] show that the algorithms are usually more effective.

In fact, if there are not any textures on the object surface, the structural feature of the object surface is quite different from that of the texture background [2]. In general, the gray variation of the object surface is much slower compared with that of the textures. So if we consider the object surface as a kind of texture, it must be a very coarse texture. It is reasonable to adopt the coarseness and directionality as the features to separate the object region from the background region.

In this paper, a new object extraction algorithm is proposed based on this idea.

2. Texture Feature : Direction Measure

There are many texture feature extraction methods[6]. The most useful texture feature is the second-order statistical feature. A new second-order statistical feature is proposed in [7], which is called direction measure (DM). There are many kinds of DMs such as first-order, second-order, and high-order DM. The high-order DM has a very high classification rate for many kinds of natural textures(its classification rate is even higher than Laws's energy measure[6])[7]. So we adopted DM as the texture feature in our algorithm.

The first-order DM is the simplest of all DMs. It can indicate coarseness and directionality of textures accurately. Its expression can be found in [7]. It contains 8 features. The coarseness is determined by the mean or median of the 8 features. The directionality of is determined by the variance of the 8 features. So in our algorithm these two simple features are adopted; mean or median of DM (MDM), and variance of DM (VDM).

$$MDM=[\sum_{i=1}^{8} d(i)]/8$$

$$VDM=[\sum_{i=1}^{8} (d(i)-MDM)]/8$$

Where { d(i), i=1,2,...,8 } is first-order DM.

The smaller the MDM, the coarser the texture is. If the texture has a large VDM, then it must have a strong directionality.

3. The Algorithm For Object Extraction

In this paper, the objects are assumed to have no textures on their surfaces. If the object surface is considered as a texture, it will be a very coarse texture. The coarseness feature is a suitable feature to distinguish the object from the texture. So, MDM is chosen as the texture feature in our algorithm. Another advantage of choosing MDM is that MDM is easy to calculate.

If a MXM window moves continuously on an NXN image in the row direction, then the image is covered with overlapping windows. If the overlapping area is chosen as maximum MX(M-1), each pixel in the image corresponds to one MXM window; The MDM is derived on each of the windows. Let the MDM be the new gray level of the pixel centered on the corresponding window, then the original image is changed to a new image which is defined here as "feature image".

Since the gray variation of the object surface is quite slow, its MDM must be very small compared with that of the texture background (if the background is made up of the same texture, the MDM is almost equal everywhere). So there are two regions in the feature image, one is the darker region corresponding to the object region, and the other is the brighter region corresponding to the background. The gray level histogram of the feature image must have a bottom "valley" between two "peaks". By choosing a proper threshold in the valley, the feature image is changed into a 2-value image. The object region is set to 1, and the background region is set to 0. The object can be extracted by multiplying the 2-value feature image with the original image.

Fig.1a is an image made up of a cloth texture and an object. The average gray level of the object surface is equal to that of the background ; So it is nearly impossible to detect the boundary between object and background by the commom edge detection algorithms. Fig.1b is the feature image obtained with the 6X6 window. Obviously, the object region is much darker than the background region in the feature image. Fig.1c is the gray level histogram of Fig.1b, and there is a sharp valley in it. The threshold is the gray value at bottom of the valley; Fig.1d is the extracted object.

If there are not any edges or corner-points in the object, the object region does not contain any bright spots or lines(we call them "gaps") in the feature image. The object can be extracted precisely like Fig.1a. But there are always some edges and corner-points on the object, and the gray variation on edges or corner-points are very sharp; So the feature image usually contains bright spots or "gaps" in the object region. In this way, the extracted object is not the same as the original one. Fig.2 shows an example. The worst is that the errors usually occur at the edges or corner-points which contain the most information of the object and are crucial for the accuracy of the object recognition.

In the feature image, the whole object region is dark except for a few points or short lines. To eliminate these "gaps", a local minimum operator is used. The definition of local minimum operator is as follows;

The gray level of one pixel(i,j) is set to the minimum of the gray levels of m pixels of the window mxm centered on (i,j).

In general, the areas of the bright sopts or lines are very small. When the local minimum operator is used, the values of the pixels in these spots or lines are replaced by the values of the neighboring pixels in the dark region. So the object region in the feature image is dark thoroughly, if the window size is properly chosen. The gray contrast of the object and background region is further enhanced; The histogram of the processed feature image will have two narrow peaks, and the threshold is easy to select. Fig.3 shows the feature image of Fig.2a after a local minimum operator is applied to Fig.2b. The conclusion is obvious and unnecessary to point out again .

4. Further Analysis of The Algorithm

The feature image is derived by calculating the MDM of each overlapping window in the original image. Each pixel in the feature image corresponds to one MXM window in the original image; Obviously, the choice of the window size M directly determines the properties of the feature image; The window should be large enough to obtain the adequate texture feature; The window should cover the area which is equal to or greater than the pseud-period of the texture. So, the coarser the background, the larger the window size is; and the finer the background, the smaller the window should be.

The coarseness of the texture background determines the window size, but the window size also affects the accuracy of extracting objects. If the window is a larger one, the range of feature variation on the boundary between object and background becomes wider, i.e. the gray variation on the boundaries between object and background regions is more smoothly in the feature image. So the 2-value feature image has position errors at the boundary between object and background(the error can not exceed M/2 pixels generally). If the object is very small, the error is very significant. The object can only be detected and can not be extracted accurately. So our algorithm is not suitable for images with very coarse texture background. In fact, this is not only the limitation of our algorithm, but all algorithms based on feature discrimination [2]. The reason is that the coarser the texture, the lower the resolution of the texture.

As a matter of fact , if the object is smaller even than the primitive of the texture background, it is also difficult for human eyes to identify it.

To eliminate the bright spots or lines in the object region in the feature image, a local minimum operator is used to process the feature image. But the local minimum operator has a disadvantage, i.e. it extends the object region by about M pixels in row and column directions. Fig.4 illustrates this phenomenen. The reason is that the gray values of the pixels of the background near the object region are replaced by the gray values of the pixels in the object region (dark region).

Fig.4 The extension of object region

To overcome this drawback, a local maximum operator is used in our algorithm. The function of the local maximum operator is just the inverse of the local minimum operator, i.e. the gray value of the central pixel in a nxn window is replaced by the maximum of the gray values of pixels in the window.

Since the bright spots or lines have been eliminated from the object region in the feature image, a local maximum operator can not make the dark region become bright again; But the average gray level of the background can be increased, i.e. the local maximum operator at least can not decrease the gray contrast between object region and background region in the feature image. Another important function of the local maximum operator is that it can reduce the area of the dark region. This is what we expect. If the window size of a local maximum operator is chosen as large as that of a local minimum operator, then the extended object region can be resumed to the exact object region.

The window size of the local minimum operator is dependent on the size of the bright spots or lines(or "gaps"); In general, the width of the gaps can not exceed the width of the window on which the texture feature is calculated. So, the window size m of local minimum operator should be a little larger than the size of that window (m>=M).

Although a local minimum operator has been used, there still are some "spoiled" points in the background region when the feature image is changed to a 2-value image. But the area of these points is usually much smaller than the objects. The last step of our algorithm is to choose a proper threshold to delete these small "spoiled" spots.

Our algorithm emphasizes the difference of coarseness features (as well as other kinds of texture features [8] [9]) between the object surface and texture background; This difference is usually much greater than the difference between the other two image textures, especially when the coarseness features of those textures don't have very large differences. So, if the background consists of different kinds of textures, the object region is still much darker than the background region in the feature image. In other words, our algorithm is still effective for those images with complex or non-stationary texture background. This is difficult or almost impossible for the modeling methods [2], because this kind of background can not be modeled by a time-unvariant linear prediction models. Fig.5a is such kind of image. Its background is composed of three kinds of txtures. Fig.5b is the extracted object; The result proves what we have discussed.

5. Experimental Results

5.1 Experimental Results of Object Extraction

To further test the efficiency of our algorithm, the authors have selected a number of natural and artificial images. These images are all 128x128 and have 256 gray levels. The average gray value of the object surface is equal or almost equal to that of the background, So the outlines of the objects can not be detected by the common edge detection algorithms; No preprocessing is made on these images.

Fig.6a is a natural picture which was taken with a common camera. Fig.6b is the extracted object. Fig.7a is a medical cell image. The detected cells are showed in Fig.7b.

Fig.8a is another medical cell image, but it was deteriorated by white noise; The ratio of signal to noise is about 5 dB. The result is not very satisfactory if our algorithm is directly applied to it. To eliminate the influence of the added noise, a local smoothing operator is used as a preprocessing procedure. Fig.8b is the feature image derived from Fig.8a , and Fig.8c is the final result.

Fig.9a is an image which is a label with two chinese words on it under the knitting cloth texture. Fig.9b is the feature image obtained with a 6x6 window. Obviously, there are very bright lines in the object region. So the local minimum operator and local maximum operator must be used. Their window size is 8x8. Fig.9c is the extracted label.

All these experiments were done on 128x128 images. The biggst position error observed does not exceed 4 pixels, and most of position errors are less than 4 pixels.

5.2 Experimental Results of Object Recognition

As a application of our algorithm, we used it to do object recognition under the complex textured background. In our experiments, the objects are all polyhedron objects, and the backgrounds are all natural textures. Of all pictures, the average gray level of the object is the same as that of the background.

The 2-D object features such as perimeter, area, perimeter/area ratio etc, are chosen in our experiments. For ten different polyhedron objects, the classification rate is about 98%. For the limit of paper volume, these experimental pictures are not listed here.

Conclusions:

From the analysis of the new algorithm an the experimental results, four conclusions can be obtained:

1) The algorithm can distinguish the boundaries between objects and background effectivly, The finer the background, the higher the resolution of the boundary. For all experiments reported here, most of the position errors are less than 4 pixels.

2) The algorithm can be adopted not only for the images with stationay texture background but also for the images with non-stationary texture background.

3) A simple smoothing method can be used as the preprocessing for the algorithm to reduce the influence of the noise.

4) Although our algorithm is based on MDM of direction measures, it can be combined with other texture features[8] [9] to form a new algorithm for object extraction.

Reference

[1] Jorge.L.C.Sanz "Fast Object Segmentation In Textured Background" Int.Conf.Proc.of ASSP 1985 pp905-908

[2] C.W.Therrien "Stochastic Model-Based Algorithm For Image Analysis" Proc.of IEEE 1986 pp532-551

[3] R.M.Haralick " Statistic and Structural Approaches To Texture" Proc. of IEEE. No 5

1979 pp768-804

[4] T.Caelli " On Perceptual Analysis Underlying Visual Texture Discrimination I" Biol.Cybern. vol.28 1978 pp167-175

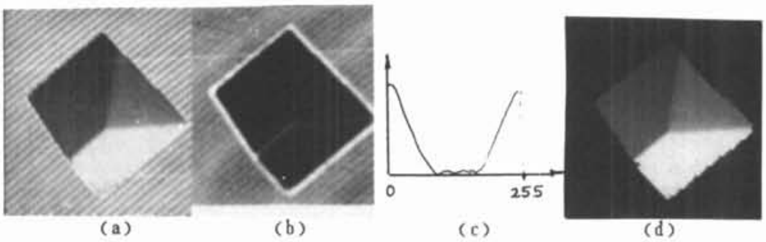
[5] J.B.Jordan "Image Segmentation Using Maximum Entropy Techniques " Int.Conf. Proc.of ASSP 1984 pp32.4.1-32.4.4

[6] K.I.Laws " Texture Energy Measures" Proc. Image Understanding Workshop nov 1979 pp47-51

[7] -------- " High-order Direction Measure for Texture Classification " Journal of Automatia 1988

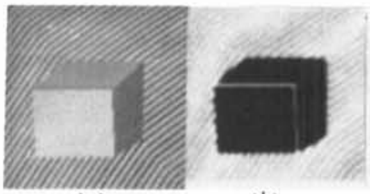
[8] A.Rosenfeld " Survey : Picture Processing 1985" Comput. Vision, Graph. and Image Processing vol.34 pp204-251 1986

[9] R.M.Haralick "Statistical and Structural Approaches to Texture " Proc. of IEEE no5 1979 pp768-804



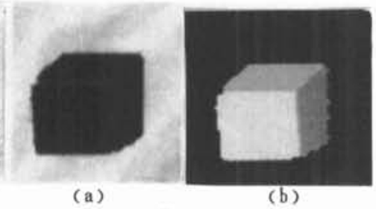
(a) (b) (c) (d)

Fig.1



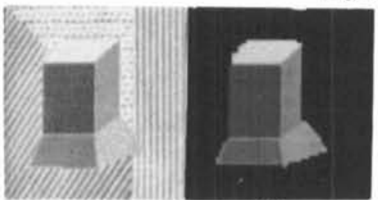
(a) (b)

Fig.2



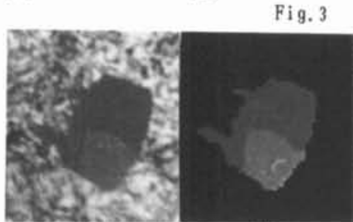
(a) (b)

Fig.3



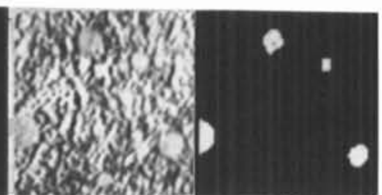
(a) (b)

Fig.5



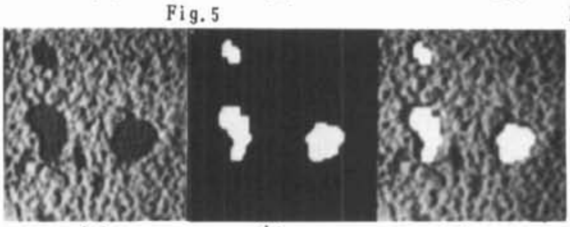
(a) (b)

Fig.6



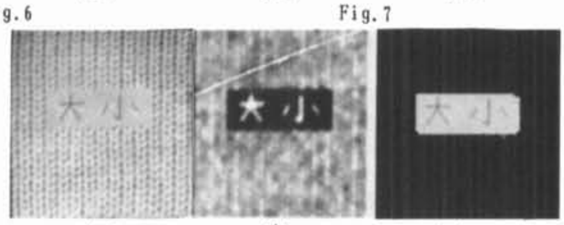
(a) (b)

Fig.7



(a) (b) (c)

Fig.8



(a) (b) (c)

Fig.9

All experimental results of objects extraction