Video Image Segmentation with Texture Gradient using DT-CWT and Spectral Clustering

Patel Shweta S.\textsuperscript{1} Prof. Kinjal Vagadia\textsuperscript{2}
\textsuperscript{1,2}Department of Electronics and Communication
Vishwakarma Govt. Engineering College Chandkheda, Ahmedabad

Abstract—A fast and automatic video object segmentation algorithm based on wavelet transform is proposed in this paper. For some applications the whole image cannot be processed directly because it is inefficient and impractical. Segmentation results in a set of images that cover the entire image. This work proposes a two stage segmentation method, which effectively process both the textured and non-textured regions. Dual Tree Complex Wavelet Transform, an extension of discrete wavelet transform, extracts texture feature from the image and orientation median filtering reduces the double edge effect at the texture edges. Watershed transform of Gaussian gradient of combined texture and non-texture feature give the first stage segmentation. The initial segmentation into super-pixels reduces computational burden and the second stage uses spectral clustering technique to cluster these primitive regions. The proposed algorithm is robust to the entire motion and local deformation of object.

Keywords: Dual Tree Complex Wavelet Transform, texture feature, watershed, super pixels, spectral clustering.

I. INTRODUCTION

Video object segmentation, which aims at the exact separation of moving objects from background, is a key technique of MPEG-4 content-based functionalities. However, even though video object segmentation has been studied for many years, it is still considered one of the most challenging issues and demands creative solutions for major breakthrough. There are mainly two kinds of images which catch the attention of researchers, namely textured and non-textured images. Texture analysis became most popular because in real life, objects and most of the images are textured in nature. Various statistical and structural approaches are available for texture analysis.

Watershed transform [11] is a powerful segmentation tool, which uses image gradient as input. Even though each regional minimum is small and insignificant, they form their own catchment basins leads to over segmentation of image. So a novel spectral clustering technique [12] is used as the second stage, which clusters the primitive region thus avoiding an excessive amount of segmentation. However, over segmentation is better than under segmentation since the former has more chances to match than later.

This paper is structured as follows. In Section II, texture watershed transform is briefly described. This section includes DTCWT for texture extraction, orientation median filtering for noise suppression, Gaussian gradient extraction and In section III the spectral clustering is explained. the section IV contain experiments and discussion. Section V contain conclusion and finally section VI contain future scope.

II. TEXTURE WATERSHED ALGORITHM

Texture watershed segmentation is a method that uses texture gradient information of input image for segmentation. Texture watershed algorithm extracts both the texture gradient and intensity gradient separately and integrates them to apply watershed segmentation.

Fig.1 shows the texture watershed algorithm. DTCWT extracted texture features undergo post processing by orientation median filtering. Texture gradient obtained from post processed information is interpolated to the size of original image. This texture gradient is integrated with intensity gradient to obtain a total gradient. The watershed transform on this total gradient gives initial segmentation. Many video segmentation algorithms have been proposed so far. They can be classified into three kinds of image segmentation based, motion segmentation based, and change detection based algorithms.

The lack of shift invariance and poor directional selectivity are two main disadvantages of real DWT [6] whereas DTCWT gives better directional selectivity and good shift invariance. The dual-tree CWT [7] consists of two parallel wavelet filter bank trees that contain carefully designed filters.

The dual-tree complex wavelet transform (CWT) is a relatively recent enhancement to the discrete wavelet transform (DWT), with important additional properties: It is nearly shift invariant and directionally selective in two and higher dimensions. It achieves this with a redundancy factor of only 2^d for d-dimensional signals, which is substantially lower than the un-decimated DWT. The multidimensional (M-D) dual-tree CWT is non-separable but is based on a computationally efficient, separable filter bank (FB).
The real 2-D dual-tree DWT of an image x is implemented using two critically-sampled separable 2-D DWTs in parallel. Then for each pair of sub-bands we take the sum and difference. The complex 2-D dual-tree DWT also gives rise to wavelets in six distinct directions. The complex 2-D dual-tree is implemented as four critically-sampled separable 2-D DWTs operating in parallel as shown in figure (7). 2-D structure needs four trees for analysis and synthesis. The pair of conjugate filters applied to two dimensional images (x, y) can be expressed as:

The complex wavelets are able to distinguish between positive and negative the diagonal sub-bands can be distinguished and horizontal and vertical sub-bands are divided giving six distinct sub-bands in each scale at orientations ±15°, ±45°, and ±75°.

The oriented and scale dependent sub-bands are visualized spatially. The magnitude of complex coefficients at level i, orientation θ denoted by |Di,θ(x,y)|.

A. TEXTURE FEATURE EXTRACTION

The wavelet transform has the ability of “Multi-resolution” analysis. Texture analysis plays an important role in the analysis of different types of images. Gabor filters [3], pyramid-structured wavelet transform [5], and tree structured wavelet transform [4] etc are used as feature extraction tools.

B. ORIENTATION MEDIAN FILTERING

When the gradient is extracted directly from wavelet sub-bands double edges are formed in the gradient magnitude. If the watershed transform is directly applied on this gradient, it results in a false narrow region along the boundary. So as a solution, a separable orientation median filtering is applied to the wavelet sub-bands before gradient extraction.

Median filter is commonly used as an edge smoothing filter of an image. A 2D median filter uses a rectangular or a square M×M window. The intensity of each pixel in the image is replaced by the median of the intensities of the points in the M×M window. Separable median filter [9] or ‘median of median’ is obtained from the successive applications of one-dimensional median filter first along the rows and then along the columns. Block diagram is given in Fig. 4. Since the wavelet sub-bands have different orientations, orientation adaptation is applied to separable median filter.

\[ S_{i,θ}(x,y) = MedFilt(\frac{π}{2})(MedFilt(θ)(D_{i,θ}(x,y))) \]

Where D(x, y) is the wavelet sub-band coefficient at level i and orientation θ. S(x, y) is the median filtered output. First apply median filter along the line normal to the sub-band orientation and then along the line parallel to that orientation. Though the output of non-separable filter is not identical with that of separable filter, performance of both in noise smoothing is very close. Separable filtering is more computationally efficient than the non-separable case.

C. TEXTURE GRADIENT COMPUTATION

Image gradient is a directional change in the intensity or color in an image, which is used to extract information from images. The gradient of a two-variable function at each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions. Here texture image is smoothened by median filtering; this removes double edge in the gradient image. The gradient operator approximation used is the commonly used Gaussian derivative function [10].

The 2D Gaussian Kernel is given as,

\[ G(x, y, σ) = e^{-\frac{(x^2+y^2)}{2σ^2}} \]

Where x and y are horizontal and vertical directions and σ is the standard deviation.

\[ \frac{∂G}{∂x}(x, y, σ) = -\frac{x}{σ^2}G(x, y, σ) \]

\[ \frac{∂G}{∂y}(x, y, σ) = -\frac{y}{σ^2}G(x, y, σ) \]

The gradient magnitude of each sub-band is, therefore given by,

\[ TG_{i,θ}(x,y) = \sqrt{(S_{i,θ}(x,y)G’x)^2 + (S_{i,θ}(x,y)G’y)^2} \]

Where is the median filtered output. G’x and G’y are the Gaussian partial derivative filters in the x and y directions and * denotes convolution.
D. INTERPOLATION

Interpolation works by using known data to estimate values at unknown points. Image interpolation works in two directions, and tries to achieve a best approximation of a pixel's color and intensity based on the values at surrounding pixels. A pixel at spatial position \((x,y)\) has one feature for each texture gradient complex wavelet sub-band defined as \(TG_i(x,y)\). Assuming a square image of dimension \(N\times N\), each sub-band at level \(l\) of the wavelet decomposition has the dimension of \(N/2^l\). However for all \(i\), the dimension of \(TG_i(x,y)\) has to be same as the image since there is the same number of features for all pixels. Therefore \(TG_i(x,y)\) is assigned with the value of the spatially related sub-band coefficient magnitude.

\[
TG_i(x,y) = \sum T_i(x,y)
\]

E. INTENSITY GRADIENT COMPUTATION

Intensity gradient of the input image, \(I(x,y)\), is also computed by using Gaussian derivative operator as for the texture gradient. Images. The gradient of a two-variable function at each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions. Here texture image is smoothened by median filtering; this removes double edge in the gradient image. The gradient operator approximation used is the commonly used Gaussian derivative function [10].

F. NORMALIZATION AND GRADIENT COMBINATION

Normalization is performed [1] to reduce the noise in gradient images. Hence, this work first finds the median of both texture and intensity gradient. The intensity gradient is normalized by four times the median intensity gradient, \(w_I\) and the texture gradient is normalized by the median of the texture gradient, \(w_T\). The four times in intensity gradient normalization is because there are sharp peaks, whereas the texture gradient is smoother. So the latter must be amplified in order to avoid suppression.

This work now combines the texture and intensity-gradient information to obtain a final gradient which is to be further segmented. The final single-valued gradient surface is computed as the combination of the texture gradient and intensity gradient.

\[
GS(x,y)\text{ is the total gradient obtained after gradient combination.}
\]

G. WATERSHED SEGMENTATION

The watershed transformation considers image gradient as a topographic surface. Pixels having highest gradient intensities correspond to watershed lines that form the region boundaries. Water placed on any pixel enclosed by a common watershed line flows downhill to a common local intensity minimum. Pixels enclosed by this common watershed line form a segment. Noise and other local irregularities in gradient image lead to over segmentation, as seen in figure., which is the well-known drawback of the watershed transform. The solution is to incorporate a preprocessing stage to limit the number of allowable regions in the image gradient. Watershed transform using immersion simulation is an efficient algorithm [12]. A concept of marker based watershed transform [8] can be used as a solution to this problem.

Here this work uses morphological H-minima transform [9] to modify the gradient surface by suppressing the gradient minima. The local gray level minima with dynamic lower than a parameter \(h\) are removed. This is done by filling the valleys in the gray value relief, until the local minimum is increased by \(h\). The H-minima transform and subsequent watershed transformation gives the modified segmented output. The output is not yet free from over segmentation. So a second stage clustering technique is used to cluster the over segmented regions.

III. SPECTRAL CLUSTERING

Spectral clustering has emerged recently as a popular clustering method that uses eigenvectors of a matrix derived from the data. Several algorithms have been proposed using the eigenvectors in slightly different ways. The set of small regions produced by over segmentation is referred to as “super-pixels”. The pre-segmentation into super-pixels reduces the computational burden, because the number of super pixels \(n\) is much less than total number of pixels in the image. So its possible to assign nodes to each region than to each pixel.

A set of data points \(x_1, x_2, \ldots, x_n\) can be represented as a similarity graph, \(G(V, A)\) whose nodes \(V=\{v_1, v_2, \ldots, v_n\}\) corresponds to the data points and edges defined through the \(n\times n\) adjacency matrix \(A\), which gives the similarity between each of sample points \(v_i\) and \(v_j\). Currently, spectral clustering algorithm can be divided into three types. Recursive spectral algorithms, such as SM algorithm [8], use the information in a single eigenvector [12]. These algorithms first divide data into two and then recursively generate more number of partitions. Multi-way spectral algorithms, such as NJW [2] algorithm, use multiple eigenvectors to directly get multiple partitions of data. Non-spectral algorithms are simple.

Clustering algorithm that clusters the data quickly. The paper explains multi-way spectral clustering can perform better and is applied here.

Given a set of block features \(\{x_1, x_2, \ldots, x_n\}\) that is to be partitioned into \(k\) clusters. Here the value of kernel parameter \(\sigma\) is set as 0.3. The algorithm is described in detail as follows.

**Step 1:** Compute affinity matrix. Affinity gives the similarity information between data points. For \(i \neq j\),

\[
A_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad \text{if} \quad i \neq j \\
A_{ij} = 0 \quad \text{otherwise}
\]
Step. 2 : Compute the diagonal matrix \( D \). Its diagonal element \((i, i)\) is the sum of each row in \( A \).

\[
D(i, i) = \sum_j A_{ij}
\]

Then construct Laplacian matrix.

\[
L = D - A
\]

Step. 3 : Compute the Eigen-values and eigenvectors of \( L \).

Step. 4 : Choose the eigenvectors corresponding to the \( k \) biggest Eigen-values and get the matrix \( U \) of size \( n \times k \).

Step. 5 : Make each row of \( U \) as a point in \( R^d \) and then use \( k \)-means (KM) to cluster them into \( k \) clusters.

Step. 6 : When the \( i \)-th row of \( U \) is assigned to the \( j \)-th cluster, the original data point \( x_i \) is then assigned to the \( j \)-th cluster.

IV. EXPERIMENTS AND DISCUSSION

In this section two more images of size 256\( \times \)256 are selected to verify this proposed method and to obtain final segmented results. In texture extraction by using DTCWT, length 10 filters have been used. The value of standard deviation \( \sigma \) in gradient detection is set as 3. These regions are meaningfully grouped into clusters using spectral clustering as seen in Table.1 shows the comparison among different methods for different input images.

V. CONCLUSION

Texture image segmentation requires proper texture gradient extraction methods. DTCWT is an effective tool for texture extraction for gradient detection. Two stages were used in this work for image segmentation. Watershed is used to perform pre-segmentation and a recently developed spectral clustering to perform the final segmentation. Spectral clustering technique used in second stage, clusters the over segmented output of watershed transform.

Though this work mainly used the texture feature, other image features such as color, intensity etc can be used for segmentation.

VI. FEATURE SCOPE

GLCM was used for feature extraction of input image for second stage. This technique improves the quality of segmentation.

REFERENCES


