Temporal Texture Recognition Model Using 3D Features

Jing Zhong and Stan Scarlaroff
Department of Computer Science
Boston University

Abstraction

Temporal textures are dynamic textures with certain spatial and temporal periodic properties. For example, the waving river, boiling water, smoke, and steam etc. We present an algorithm that considers each temporal texture as a 3D volume data in a spatiotemporal space, and extract the features from this 3D volume data. Different from the previous approaches, the features are derived from 2nd order surface features (such as curvature) of the 3D surface in spatiotemporal space, while the feature distribution of each temporal texture is modeled by a Gaussian Mixture Model. The classification is realized by Baysian maximum a posterior (MA) rule. Our experiments showed that 98.6% recognition accuracy is achieved for 9 temporal textures.

1. Introduction

Temporal texture is the image sequence exhibiting certain spatial or temporal periodic structure, for example, the boiling water, the waving river, the steams and so on. Classification and recognition texture has very wide applications, such as object segmentation, novelty detection, inverse rendering etc.

Recognition of temporal texture has been investigated by many researchers. M. Szummer [1] presented a method to model temporal texture using a spatiotemporal autoregressive model. K. Otsuka [2] extracted the temporal and spatial features based on the tangent plane distribution in the spatiotemporal space. The tangent plane distribution is obtained using 3D Hough Transform from motion trajectory surface computed by the contour surfaces in the image sequence. Because the motion trajectory surfaces are built on the binary surface images, quantization noise may be introduced due to binarization. Furthermore, they do not consider the second order surface features, such as curvatures of the surface. This may result in the loss of some useful motion features, such as acceleration.

In this report we attempt to improve the temporal texture recognition algorithm by the following aspects: First, to deal with the limitations of the 1st order surface features, we applied the 2nd order surface features, which are extracted from the curvatures on the trajectory isosurface generated by the temporal texture. 3D edge detection is first conducted to extract likely trajectory points in the 3D surface, afterwards each feature vector is extracted in the 3d cubic voxel containing non-zero edges. These features include the tangent plane direction, edge strength and the principle curvatures. Notice that these features are extracted from intensity volume data instead of binary data, thus it would avoid the loss of information and quantization noise due to binarization. Second, we use a Mixture of Gaussian (GMM) method to model the distribution of the features in the feature space. This modeling process makes the system be able to capture the
variations of the feature vectors in the different spatiotemporal locations for one single temporal texture. The limitation of the classic GMM is its requirement for a fixed number of mixtures, which have to be determined a prior. This may be not be suited for temporal texture because the number of the modes of the density function is often unknown a prior. To handle this problem, the Greedy Gaussian Mixture Model [4] is employed such that the algorithm is able to build the model in an adaptive manner, that is, the mixture component number is decided adaptively using Greedy Gaussian Mixture Models.

The report is organized as following: Section 2 gives the representation of the temporal texture, i.e. the feature vector extraction. Section 3 gives the GMM modeling of the probability density function. Section 4 presents the experiments and discussion.

2. Representation of Temporal Texture

The image sequence of temporal texture is modeled as a time space volume data, where t axis become z axis in this modeling, and x, y axes keep unchanged. Formally, this defines a three dimensional function $I = f(x,y,t)$, where $x, y$ are the coordinates of each spatial space and $t$ is the coordinate of the temporal space, $I$ is the intensity value, which ranges from 0 to 255.

2.1 3D Gradient Filter

Firstly, a 3D gradient filter is applied to extract the 3D edges at every 3D image point. We applied the algorithm proposed by M. Brejil and M.Sonka [3] to extract the 3D edges. Their 3D edge detector is based on fitting a tri-cubic polynomial into the original volume data in the least square error sense. Their methods have the following advantages, 1) it is applied to image data directly, without the need of binarization 2) it models the integrative character of data acquisition, 3) its computation complexity is as expensive as any other convolution-based directional edge detector. Based on the gradient vector, we can set a threshold to obtain the edge surfaces in the 3D spatiotemporal space. This is different from the method in [2], where the edge surface is obtained by stacking edge binary images. This method would improve the accuracy of edge surface extraction by incorporating the information from the neighborhood frames.
The examples of 3D edge surfaces are shown in Figure 2.

Figure 2. The examples of the 3D edge surfaces, which are represented by images in \(xy\), \(xt\), and \(yt\) space respectively.
2.2 Representation of temporal texture in spatiotemporal space

After obtaining the 3D edge surfaces, the feature vectors are extracted on these surfaces, including the gradient vector of the 3D edge and the maximum principle curvature.

2.2.1 Feature vector

The feature vector will be extracted only on the edge voxels composing the 3D edge surfaces. The gradient vectors with its strength and direction, and the maximum principle curvature are combined to form a feature vector.

The gradient vector of each edge voxel is computed by the 3D edge filter. The gradient vectors are normalized to surface norm. And the directions of the surface norm are extracted as three component of the overall feature vector.

The algorithm for maximum principle curvature is proposed by computing the curvatures at the locations designated as edge points using the partial derivatives of the image directly [5]. This could avoid the problem, establishing the link between 3D edge detection and local surface approximation, within the classical approaches that use local surface fitting. The maximum principle curvature at a voxel is depended on the second partial derivatives computed as Hessian Matrix.

2.2.2 Volume Data separation

The original video spatiotemporal volume data is further separated into a number of small cubes, shown in the Figure 3. The mean of the feature vector is computed within each small cube. Finally, the temporal texture in spatiotemporal space is represented by the mean vectors.

Figure 3.
3. Training and classification

The distribution density function of the feature vectors of each temporal texture is shaped by the Gaussian Mixture Model, which can be expressed as the following form:

\[ p(O \mid T) = \sum_{j=1}^{M} p(O \mid j)p(j) \]

\[ p(O \mid j) = \frac{1}{2\pi j} \exp \left( -\frac{1}{2} (O - \mu_j)^T \Sigma_j^{-1} (O - \mu_j) \right) \tag{1} \]

where the O is the feature vector, the T is the index of the temporal textures, M is the number of the mixture Gaussian functions, and the j is the index of the mixture component. The training of the GMM on a set of training data can be realized by Expectation Maximization [6]. This model has its limitation of requiring the knowledge of the mixture number M, which is usually unable to be determined a prior. Here we use a method that is developed by [4]. In [4], a greedy learning algorithm is utilized. The greedy algorithm split the mixture component when the additional mixture is needed based on certain criterion.

The classification of the temporal textures are achieved by the Maximum A Posterior (MAP) classification using the Bayes rules:

\[ T = \arg \max \log(p(T \mid O)) \]

\[ = \arg \max \left[ \log p(O \mid T) + \log p(T) \right] \tag{2} \]

Since we assume that the prior is uniform over all temporal texture, the above classification procedure only involves in the comparison of the observation likelihoods of different temporal texture.

4. Experiments

We evaluate the algorithms on 9 temporal textures. The data come from M. Szummer’s standard temporal texture database [9]. The data consists of a diverse set of temporal textures including boiling water, escalator, fire, flags, fountain, river, smoke, steam and trees. We divide each dataset into a number of small cubes. The size of the cubes is 15x15x5, x, y and t respectively. Our features include: mean of gradient direction (3d) and mean of maximum curvature (1d) in each small cube. We use a Leave-One-Out cross validation to test the algorithm: the dataset is divided into 8 subsets, half in x, y and t direction respectively, afterwards the algorithm is trained using 7 subsets while tested in the remaining one. An 8x8 confusion matrix then is obtained to show the performance. Each line and column of the confusion matrix represents one temporal texture.

We evaluated the algorithm under two experimental scenarios. In the first scenario, those small cubes without 3D edge are included in training and testing, but the feature
vectors are set to [0, 0, 0, 0]. In the second scenario, the cubes without edges are excluded in training and testing. The Confusion Matrixes corresponding these two testing scenarios are shown in Figure 4. Based on the testing results, we can conclude that those cubes without edges should be eliminated before the testing. We can come up with the recognition accuracy 98.6% for the first confusion matrix, and the recognition accuracy for the second confusion matrix rate is 89.9%.

(a) Confusion Matrix deleting the cubes without edge

(b) Confusion Matrix not deleting the cubes without edge

Figure 4. The Confusion Matrix

5. Discussions and Future Work

A 2\textsuperscript{nd} order feature based classification method is presented in the paper. These features improve the representation accuracy of the temporal texture. Thus, this algorithm is able to increase the recognition performance. Furthermore, a Baysian classification method with mixture models is applied to take account for the uncertainty and noisy nature of the feature extraction procedure. The experimental results showed the algorithm recognizes the temporal textures with very high accuracy. Future work is to replace the Guassian mixture model with the Markov Random Field, so that the model can incorporate the spatial relationship information of those 3D voxels.
6. Reference


