Motion Segmentation Based on a New Genetic Algorithm

Yuzhen Li, Mingying Fan, Ghoneim Mohamed, Jianming Lu, Hiroo Sekiya and Takashi Yahagi

Graduate School of Science and Technology, Chiba University
1-33 Yayoi-cho, Inage-ku, Chiba, 263-8522, Japan
Email: yuzhenli@graduate.chiba-u.jp

1. Introduction

Video surveillance has been used in many monitoring security sensitive areas such as banks, department stores, highways, crowded public places and borders. Detecting moving regions such as vehicles and people is the first basic step of almost every vision system, because it provides a focus of attention and simplifies the processing on subsequent analysis steps. It is one of the most difficult tasks, the motion segmentation accuracy determines the eventual success or failure of computerized analysis procedures. For this reason considerable care should be taken to improve the probability of rugged segmentation.

Many approaches exist for detecting moving object in a sequence of images. Commonly used techniques for motion detection are background subtraction, temporal differencing and optical flow. Background subtraction\cite{1,2,3,4,5,6} attempts to detect moving regions in an image by subtracting current image from a reference background image in a pixel-by-pixel fashion. Temporal differencing\cite{7,8} makes use of pixel intensity difference between two or three consecutive frames in an image sequence to extract moving regions. Motion segmentation based on optical flow\cite{9,10} uses characteristics of flow vectors of moving objects over time to detect change regions in an image sequence. In these methods, background subtraction provides the most complete feature data, but is extremely sensitive to dynamic scene changes due to lighting and extraneous events. Temporal differencing is very adaptive to dynamic environments, but generally has a poor performance in extracting all relevant feature pixels. Optical flow can be used to detect independently moving objects in the presence of camera motion. However, most optical flow computation methods are computationally complex, and cannot be applied to full-frame video streams in real-time without specialized hardware.

Recently, genetic algorithms based motion segmentation have been proposed\cite{11}. It proposes a new video sequence segmentation method based on the genetic algorithm (GA) that can improve computational efficiency. The computation is distributed into chromosomes that evolve using distributed genetic algorithms (DGAs). But this method needs the segmentation in spatial and temporal field separately. Then combines them together, which is time consuming.

Inspired by the paper \cite{11}, we introduce a new motion segmentation method which combines the spatial segmentation and temporal segmentation together. First we construct the image model based on the Markov Random Fields (MRF) for each frame. Then the segmentation is represented by the minimization of a posterior energy function. We use the genetic algorithm (GA) to find the solutions. The background differing and evolution probability are combined to find the unstable individuals. The advantage of this method is that it decrease the number of the evolution individuals and decrease the computation time consuming.

2. Image Model

Let \( R \) be the 3-D volume of \( M \times N \) lattices, such that an element \( R_{rt} \) indexes a pixel at site \( r \) and time \( t \). Let \( G \) be the input sequence of color images defined on \( R \). \( G \) is considered as degraded by i.i.d (independent identically distributed) zero-mean random noise \( N \) is defined on \( R \). \( B = \{B_{rt} \mid B_{rt} \in \Lambda, 1 \leq r \leq M, 1 \leq t \leq N \} \) represents the label configurations defined on \( R \), wherein \( B_{rt} \) is a discrete random variable taking a value in the label set \( \Lambda = \{ \lambda_1, \cdots, \lambda_R \} \). A spatiotemporal neighborhood in \( R \) is defined as \( \Gamma = \{ \eta_{rt} \} \), where \( \eta_{rt} \) is the set of neighboring sites \( (r,t) \). Then, \( B \) is modeled by a spatio-temporal MRF based on \( R \) with respect to \( \Gamma \), because it satisfies the following spatiotemporal Markovian property\cite{11}.

\[
P(B_{rt} = b_{rt} \mid B_{ij} = b_{ij}, (r,t) \neq (i,j)) = P(B_{rt} = b_{rt} \mid B_{ij} = b_{ij}, (i,j) \in \eta_{rt})
\]

(1)

Let \( \omega \) be a realization of \( B \). Then, the segmentation is to estimate the label configuration \( \omega \) that maximizes the following posterior distribution for a fixed input video \( g \).

\[
arg\max_{\omega} P(B = \omega \mid G = g) = \arg\max_{\omega} \frac{P(g \mid \omega)P(\omega)}{P(g)}
\]

(2)

\( P(g) \) can be discarded since it is a constant. Accordingly, the posterior distribution is affected by the prior probability \( P(\omega) \) and the likelihood function \( P(g \mid \omega) \), which are represented...
as follows:

\[ P(g \mid \omega) = P(N = g - M(\omega) \mid \omega) \]
\[ = \Pi_{(r,t) \in R} P(n_{rt} = g_{rt} - M(\omega_{rt}) \mid \omega_{rt}) \]
\[ = \Pi_{(r,t) \in R} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(g_{rt} - M(\omega_{rt}))^2}{2\sigma^2}\right] \]  
\[ (3) \]

\[ P(\omega) = \exp(-U(\omega)) = \exp\left[-\sum_{c \in C} \{S_c(\omega) + T_c(\omega)\}\right] \]  
\[ (4) \]

\( M \) is the function that the label of a pixel corresponds to the estimated color vector and \( \sigma \) is the noise variance. \( C \) is a possible set of cliques. The energy function \( U(\omega) \) is obtained by the summation of spatial potentials \( S_c(\omega) \) and temporal potentials \( T_c(\omega) \) over all possible cliques.

Then, formula (2) can be represented by minimization of a posterior energy function as follows:

\[ \text{argmin}_\omega \left\{ \sum_{c \in C} \{S_c(\omega) + T_c(\omega)\} + \sum_{(r,t) \in S} \left(\frac{g_{rt} - M(\omega_{rt}))^2}{2\sigma^2} + \frac{1}{2}\log(2\pi\sigma^2)\right) \right\} \]  
\[ (5) \]

Let \( r_{rt} \) represents a set of cliques containing pixel \((r,t)\). Since \( C \) is equal to the sum of \( r_{rt} \) for all pixels, Equation (5) can be represented by:

\[ \text{argmin}_\omega \sum_{(r,t) \in S} \sum_{c \in r_{rt}} \left\{ (S_c(\omega_{rt}) + T_c(\omega_{rt})) + \frac{(g_{rt} - M(\omega_{rt}))^2}{2\sigma^2} + \frac{1}{2}\log(2\pi\sigma^2) \right\} \]  
\[ (6) \]

3. Motion segmentation algorithm

Motion segmentation is formulated as maximizing the posterior distribution. This maximization is carried out by individuals that evolve using Distributed Genetic Algorithm (DGA). First, we overview the GA algorithm.

3.1. Genetic algorithm overview

A genetic algorithm (GA) is a robust stochastic search technique used in computing to find the solutions to optimization and search problems. GAs have been employed with success in a variety of problems such as combinatorial optimization, system identification, image enhancement and so on. The lure of GAs is that they do not require differentiability or even continuity of the search space, it can decrease the time cost according to parallel computation and avoid the local optima.

A typical genetic algorithm requires two things to be defined: (1) A genetic representation of the solution domain, (2) A fitness function to evaluate the solution domain.

A standard representation of the solution is an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, that facilitates simple crossover operation. Variable length representations were also used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and free-form representations are explored in Human-based Genetic Algorithm (HBGA).

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem we want to maximize the total value of objects that we can put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all objects in the knapsack if the representation is valid, or 0 otherwise. In some problems, it is hard or even impossible to define the fitness expression; in these cases, interactive genetic algorithms are used.

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, then improve it through repetitive application of mutation, crossover, and selection operators.

3.2. Apply GA to motion segmentation

3.2.1. Chromosome

According to the analysis above, to apply GAs to motion segmentation problem, an appropriate structural representation of the candidate solution is needed. The representation of a solution is an important choice in the algorithm because it determines the data structures that will be modified in the crossover and mutation operators. A chromosome consists of a label and RGB color vectors at pixel \((m,n)\), and is located at pixel \((m,n)\). The former is used as the region number for the pixel where the chromosome is located, and the latter is used to assign a fitness value to the chromosome.

Application of the GA to a motion segmentation problem requires the definition of a fitness function. In this paper, the fitness of a chromosome is defined as \(-U_{rt}\). The higher the fitness value, the better the solution. Thus, to maximize its fitness value, each individual is evolved by iteratively performing GA operators. In a DGA, the operators function on
the neighbors of an individual rather than the whole population. Here, the neighbors of an individual are composed of the individuals located within a \( w \times w \) window centered on the pixel \((r, t)\).

Each individual has two values: the differing value \((D_t(r))\) and the evolution probability value \((PE_{rt})\). Based on the two value, the individuals are classified as either stable or unstable individuals. An individual is categorized as unstable if the following condition is satisfied:

\[
D_t(r) = |I_t(r) - B_t(r)| > T_t
\]

\[
\text{or } PE_{rt} \geq \frac{1}{2}(1 - C_r(\text{or} M_r))
\]

Here, \(I_t(r)\) represents the intensity value of pixel at position \(r\), at time \(t\), and \(B_t(r)\) represents the current background intensity value of pixel \(r\), learned by observation over time. \(T_t(r)\) is a threshold describing a statistically significant intensity change of pixel at position \(r\). The difference between the intensity value in the background and the intensity value in the current frame represents the change of the intensity.

The probability of an individual \(C_r\) is denoted as \(PE_{rt}\), and defined as follows:

\[
K = \max\{\Delta U(0, t), \cdots, \Delta U(M_1 M_2 - M_1 - M_2 + 1, t)\}
\]

\[
PE_{rt} = \frac{\Delta U(r, t)}{K}
\]

where \(\Delta U(r, t) = |U(r, t) - U(r, t - 1)|\) is the local energy variance of individual \(C_{r}\) at time \(t\). The evolution probability of a individuals is directly proportional to the variance of its local energy.

If only the \(PE_{rt}\) is used to find the unstable individuals, we could not get the entire individuals. For example, if the moving object moving very slow, some moving intensity fixels in the moving object could not chang very apparently or even not changed. Combining the background subtraction with the evolution probability, we could get much better segmentation results.

3.2.2. Initialization

In our method, the individuals mapped to the first frame are initialized with totally random values, and all of them have the same evolution probabilities. Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). The advantage of our method is that the individuals of the subsequent frames are initiated from the segmentation results of the previous frame, and then classified into stable and unstable ones according to the condition in formula (7). Stable individuals are evolved by the only selection, whereas the unstable individuals are evolved using all the operators for each generation. It could decrease the computation complexity.

3.2.3. Selection

During each successive epoch, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, the individual with the highest fitness in the neighborhood is selected.

3.2.4. Reproduction

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover and mutation. Crossover is a variant of uniform crossover. First, a neighboring individual is randomly picked in the window. Then, one component in the feature vector is chosen and replaced by the corresponding value of the feature vector in the current individual, thereby yielding two new individuals, one of which is then selected. For mutation, a random position is chosen along the feature vector, then the corresponding value is added to the value sampled from the normal distribution.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each child, and the process continues until a new population of solutions of appropriate size is generated.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

3.2.5. Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are: (1) A solution is found that satisfies minimization criteria; (2) Fixed number of generations reached; (3) Allocated budget (computation time/money) reached; (4) The highest ranking solution’s fitness is reaching or has reached a plateau such
that successive iterations no longer produce better results;

The stability criterion is reached if the stability is above a stability threshold. The stopping criterion is reached when the stability is above the maximal stability, or when the frequency reaching to the stability criterion is above a predefined number (stability number). The stopping criterion is also reached when the number of generations is higher than the maximal number.

4. Experimental results

We present segmentation results on the 33th frame of the video sequence. Figure 1(a) shows the background frame of the original sequence, Figure 1(b) shows the 33th frame. Figure 2 is the segmentation result. From the result, we see that the moving object could be segmented very well. For future work, I have to do the performance comparison in the number of generations, posteriori energy function value and the computation time with other methods; Use the MRF model to estimate the motion and track the moving object temporally.

Figure 1: Original frame

Figure 2: Segmentation result

5. Conclusion

This paper introduce a new segmentation method based on Markov random field model which uses the motion information by using the GAs algorithm. It reduces the computational time and improves the quality of the segmentation results by combining the background substraction and evolution probability to divide the chromosomes to unstable ones in theory.

References


